

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

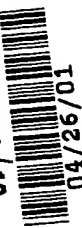
- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images,
Please do not report the images to the
Image Problem Mailbox.



JC973 U.S. PTO
09/841875



대한민국 특허청

KOREAN INDUSTRIAL
PROPERTY OFFICE

별첨 사본은 아래 출원의 원본과 동일함을 증명함.

This is to certify that the following application annexed hereto
is a true copy from the records of the Korean Industrial
Property Office.

출원 번호 :
Application Number

특허출원 2000년 제 38161 호

출원 년 월 일 :
Date of Application

2000년 07월 05일

출원인 :
Applicant(s)

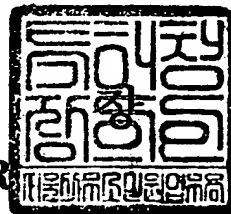
주식회사 에이디칩스



2001 02 19
 년 월 일

특 허 청

COMMISSIONER



출력 일자: 2001/2/22

【서류명】	출원인정보변경 (경정)신고서
【수신처】	특허청장
【제출일자】	20010207
【출원인】	
【명칭】	주식회사 에이디칩스
【출원인코드】	119986031478
【대리인】	
【성명】	황선웅
【대리인코드】	919990004971
【변경사항】	
【경정항목】	한글 성명(명칭)
【경정전】	주식회사 아시아디자인
【경정후】	주식회사 에이디칩스
【변경사항】	
【경정항목】	영문 성명(명칭)
【경정전】	Asia Design Co., Ltd.
【경정후】	Advanced Digital Chips Inc.
【취지】	특허법시행규칙 제9조·실용신안법시행규칙 제12조·의장법 시행규칙 제28조 및 상표법시행규칙 제23조의 규정에 의하 여 위와 같이 신고합니다.

【서류명】	특허출원서
【권리구분】	특허
【수신처】	특허청장
【참조번호】	0001
【제출일자】	2000.07.05
【발명의 명칭】	프로그램 테스트 및 디버깅이 용이한 중앙처리장치
【발명의 영문명칭】	Central processing unit for easily testing and debugging of program
【출원인】	
【명칭】	주식회사 아시아디자인
【출원인코드】	1-1998-603147-8
【대리인】	
【성명】	황선웅
【대리인코드】	9-1999-000497-1
【포괄위임등록번호】	1999-068997-5
【발명자】	
【성명의 국문표기】	조경연
【성명의 영문표기】	CH0,Kyung Youn
【주민등록번호】	550219-1063314
【우편번호】	612-030
【주소】	부산광역시 해운대구 좌동 1271번지 신성아파트 104-1402
【국적】	KR
【발명자】	
【성명의 국문표기】	임종윤
【성명의 영문표기】	LIM,Jong Yoon
【주민등록번호】	701001-1030554
【우편번호】	132-021
【주소】	서울특별시 도봉구 방학1동 651-35
【국적】	KR
【발명자】	
【성명의 국문표기】	이근택
【성명의 영문표기】	LEE,Geun Taek
【주민등록번호】	740219-1394417

【우편번호】	120-121
【주소】	서울특별시 서대문구 남가좌1동 124-193
【국적】	KR
【발명자】	
【성명의 국문표기】	한상서
【성명의 영문표기】	HAN,Sang Seo
【주민등록번호】	721110-1560411
【우편번호】	407-052
【주소】	인천광역시 계양구 계산2동 영동아파트 1동 310호
【국적】	KR
【발명자】	
【성명의 국문표기】	민병권
【성명의 영문표기】	MIN,Byung Gueon
【주민등록번호】	631005-1005827
【우편번호】	464-860
【주소】	경기도 광주군 초월면 대쌍령리 61-1 우림아파트 107동 205호
【국적】	KR
【발명자】	
【성명의 국문표기】	이희
【성명의 영문표기】	LEE,Heui
【주민등록번호】	580701-1148514
【우편번호】	463-020
【주소】	경기도 성남시 분당구 수내동 55번지 파크타운 롯데아파트 137-1206
【국적】	KR
【심사청구】	청구
【취지】	특허법 제42조의 규정에 의한 출원, 특허법 제60조의 규정에 의한 출원심사를 청구합니다. 대리인 황선웅 (인)
【수수료】	
【기본출원료】	20 면 29,000 원
【가산출원료】	24 면 24,000 원
【우선권주장료】	0 건 0 원

【심사청구료】	34	항	1,197,000	원
【합계】	1,250,000		원	
【감면사유】	중소기업			
【감면후 수수료】	625,000		원	
【첨부서류】	1.	요약서·명세서(도면)_1통		

【요약서】**【요약】**

본 발명의 프로그램 테스트 및 디버깅이 용이한 중앙처리장치는 호스트 컴퓨터와 데이터 통신을 위한 데이터 통신부; 중앙처리장치의 동작모드가 일반 동작 상태를 나타내는 일반 동작 모드인지 디버깅 상태를 나타내는 디버깅 모드인지를 나타내는 1 이상의 플래그들을 구비한 상태 레지스터; 메모리상의 디버깅 스택 영역을 지정하는 디버깅 스택 포인터 레지스터; 브레이크 레지스터에 저장된 값과 중지데이터를 비교하여 동일하면 중앙처리장치를 디버깅 모드로 진입시키는 비교부를 구비하여 상기의 브레이크 레지스터에 저장된 값과 중지 데이터가 일치하면 상기의 중앙처리장치를 디버깅 모드로 전환시키고 상기의 1이상의 플래그들이 디버깅 모드를 나타내는 값을 가지며 상기의 디버깅 스택 포인터 레지스터를 디버깅 프로그램의 데이터를 저장하는 메모리를 지정하는 스택 포인터로 사용하며 데이터 통신수단을 이용하여 호스트 컴퓨터로부터의 명령에 따라 디버깅을 수행하는 것을 특징으로 한다.

【대표도】

도 3

【색인어】

디버깅, 호스트 컴퓨터, 인서킷 에뮬레이션

【명세서】**【발명의 명칭】**

프로그램 테스트 및 디버깅이 용이한 중앙처리장치{Central processing unit for easily testing and debugging of program}

【도면의 간단한 설명】

도 1은 종래의 인서킷 에뮬레이터를 사용하여 프로그램의 테스트 및 디버깅을 하기 위한 블럭다이어그램,

도 2는 종래의 중앙처리장치 내에 직렬통신장치와 백그라운드 디버깅 모니터부가 내장되어 프로그램의 테스트 및 디버깅을 하기 위한 블럭다이어그램,

도 3은 본 발명의 프로그램 테스트 및 디버깅이 용이한 중앙처리장치의 블럭다이어그램,

도 4는 본 발명의 프로그램 테스트 및 디버깅을 위한 상태 레지스터의 구성도,

도 5a는 일반 동작을 위한 응용 프로그램 및 디버깅 프로그램이 동일한 메모리에 저장된 경우 벡터 처리의 동작도,

도 5b는 디버깅 프로그램이 일반 동작을 위한 응용 프로그램과는 다른 독립된 메모리에 저장된 경우 벡터 처리의 동작도,

도 5c는 디버깅 프로그램이 일반 동작을 위한 응용 프로그램이 저장된 메모리 및 다른 독립된 메모리에 모두 저장된 경우 벡터 처리의 동작도,

도 6은 본 발명의 프로그램 테스트 및 디버깅 방법을 위한 순서도,

도 7은 본 발명의 프로그램 테스트 및 디버깅 방법의 다른 실시례의 순서도 이다.

【발명의 상세한 설명】**【발명의 목적】****【발명이 속하는 기술분야 및 그 분야의 종래기술】**

<10> 본 발명은 프로그램 테스트 및 디버깅이 용이한 중앙처리장치 및 프로그램 테스트 및 디버깅 방법에 관한 것으로, 특히 디버깅 상태에서 호스트 컴퓨터로부터 다운로드 된 테스트 및 디버깅을 하기 위한 응용 프로그램을 용이하게 테스트 및 디버깅을 수행할 수 있는 프로그램 테스트 및 디버깅이 용이한 중앙처리장치에 관한 것이다..

<11> 일반적으로 중앙처리장치(Micro-Processor)는 메모리에 저장된 다수의 명령어들의 집합으로 이루어진 프로그램을 읽어 들여 이를 명령어 해석기에 의해 해석하고, 실행기에 의해 각 명령어에 해당하는 동작을 실행한다.

<12> 메모리에 저장된 다수의 명령어들의 집합으로 이루어진 프로그램이 올바르게 작성되었는지를 테스트하고, 프로그램이 올바르게 작성되어 있지 않은 경우 프로그램을 디버깅(Debugging)하여야 한다.

<13> 도 1은 종래의 인서킷 에뮬레이터를 사용하여 프로그램의 테스트 및 디버깅을 하기 위한 한 예의 블럭다이아그램이다.

<14> 도 1의 인서킷 에뮬레이터를 사용하여 프로그램의 테스트 및 디버깅을 하기 위한 블럭다이아그램은 중앙처리장치, 입출력장치(1) 및 메모리(2)로 구성되는 대상 시스템(4), 대상 시스템(4)을 테스트 및 디버깅하기 위해 대상 시스템(4)의 중앙처리장치를 대체할 수 있는 회로로 구성되어 대상 시스템(4)의 중앙처리장치를 중앙처리장치 모듈(Module)(3)로 대체시켜 중앙처리장치 모듈(3)과 연결된 인서킷 에뮬레이터(In Circuit

Emulator:ICE)(5) 및 디버깅을 하기 위한 응용 프로그램의 테스트 및 디버깅을 위한 명령 등을 인서킷 에뮬레이터로 출력시키는 호스트 컴퓨터(6)로 구성된다.

<15> 인서킷 에뮬레이터(5)는 호스트 컴퓨터(6)로부터 응용 프로그램을 입력받아 응용 프로그램을 디버깅할 수 있는 디버깅용 프로그램을 저장하고 있으며, 호스트 컴퓨터(6)로부터 입력되는 명령에 따라 내부의 중앙처리장치를 동작시킨다. 그리고 제어 버스, 어드레스 버스 및 데이터 버스를 직접적으로 제어하여 대상 시스템(4)의 입출력장치(2) 및 메모리(3)에 필요한 입출력을 수행할 수 있고, 중앙처리장치의 내부 레지스터를 읽거나 내부 레지스터의 데이터를 변경할 수 있다.

<16> 즉, 인서킷 에뮬레이터(5)는 호스트 컴퓨터(6)로부터 디버깅을 하기 위한 응용 프로그램을 입력받아 프로그램을 수행시켜 인서킷 에뮬레이터(5) 내의 중앙처리장치를 동작시킨다. 그리고 제어 버스, 어드레스 버스 및 데이터 버스를 직접적으로 제어하여 대상 시스템(4)의 입출력장치(2) 및 메모리(3)에 필요한 입출력을 수행하고, 응용 프로그램의 테스트 및 디버깅을 위하여 강제적으로 응용 프로그램의 수행을 중지시킨 후 호스트 컴퓨터(6)로부터의 명령에 따라 대상 시스템(4)의 입출력장치(2) 및 메모리(3)의 데이터를 읽거나 쓰고 또는 내부 레지스터에 저장된 데이터 값을 읽거나 써서 응용 프로그램이 올바르게 작성되었는지를 판단하고, 오류가 있는 경우 응용 프로그램을 디버깅한다.

<17> 상기 종래의 인서킷 에뮬레이터를 사용하여 응용 프로그램의 테스트 및 디버깅을 하는 경우 반드시 대상 시스템의 중앙처리장치를 인서킷 에뮬레이터에 연결된 중앙처리장치 모듈로 대체될 수 있는 경우에만 유용하게 사용할 수 있고, 인서킷 에뮬레이터내에는 대상 시스템의 중앙처리장치의 기능을 하는 하드웨어가 있어야 하므로 인서킷 에뮬레

이터는 고가의 장비라는 문제점을 가지고 있다. 그리고 대상 시스템을 하나의 반도체 칩으로 집적화 시킨 경우 중앙처리장치(3)를 대체할 수 없으므로 인서킷 에뮬레이터를 사용할 수 없는 문제점을 가지고 있다.

<18> 도 2는 종래의 중앙처리장치 내에 백그라운드 디버깅 모니터부가 내장되어 프로그램의 테스트 및 디버깅을 하기 위한 블럭다이어그램이다.

<19> 도 2의 블럭다이어그램은 도 1의 대상 시스템을 하나의 반도체 칩으로 집적시킨 경우 인서킷 에뮬레이터를 사용할 수 없는 문제점을 해결하기 위해 중앙처리장치는 백그라운드 디버깅 모니터부(Background Debugging Monitor : BDM)와 직렬 통신부를 별도로 가지고 있다.

<20> 도 2의 종래의 중앙처리장치 내에 백그라운드 디버깅 모니터부가 내장되어 프로그램의 테스트 및 디버깅을 하기 위한 블럭다이어그램은 별도의 백그라운드 디버깅 모니터부(11) 및 직렬 통신부(12)를 가진 중앙처리장치(13), 입출력장치(14) 및 메모리(15)가 하나의 반도체 칩으로 집적화된 대상 시스템(10)과 직렬 통신부(12)를 통해 응용 프로그램의 테스트 및 디버깅을 하기 위한 호스트 컴퓨터(20)로 구성된다.

<21> 백그라운드 디버깅 모니터부(11)는 호스트 컴퓨터(20)로부터 직렬 통신부(12)를 통해 입력된 명령어를 해석하여 필요한 동작을 수행하기 위해 자체적으로 레지스터, 제어유니트 등으로 구성된 콘트롤러로 중앙처리장치(13) 내부의 모든 구성요소들을 직접적으로 제어할 수 있어야 한다. 즉, 백그라운드 디버깅 모니터부(11)는 내부 버스에 연결되어 레지스터를 읽거나 값을 변경할 수 있다. 그리고 제어부, 메모리 어드레스 레지스터 및 메모리 데이터 레지스터를 제어하여 제어신호버스, 어드레스 버스 및 데이터 버스를 제어한다.

<22> 도 2의 종래의 백그라운드 디버깅 모니터부가 내장된 중앙처리장치는 직렬 통신부(12)를 통하여 호스트 컴퓨터(20)와 백그라운드 디버깅 모니터부(11)의 통신이 가능해야 하는데, 이에 요구되어지는 호스트 컴퓨터(20)와 중앙처리장치(13)를 연결하기 위한 입출력선은 2~3개의 선이 필요하므로 도 1에서와 같이 중앙처리장치 전부를 대체하는 경우 보다 효율적이다.

<23> 종래의 백그라운드 디버깅 모니터부는 중앙처리장치(13) 내부의 모든 구성요소들을 직접적으로 제어하여야 하므로 도 2의 백그라운드 디버깅 모니터부가 내장된 중앙처리장치는 고가이고, 복잡한 하드웨어를 가지는 문제점을 가지고 있다.

【발명이 이루고자 하는 기술적 과제】

<24> 본 발명의 목적은 디버깅 입력신호를 활성화시켜 사용자 모드를 갖는 중앙처리장치의 상태에 디버깅 모드를 새로이 추가하고, 디버깅 모드에서 호스트 컴퓨터로부터 다운로드 된 테스트 및 디버깅을 하기 위한 응용 프로그램을 용이하게 테스트 및 디버깅함으로써 디버깅 모듈을 포함한 중앙처리장치의 하드웨어가 단순하게 구성되어 중앙처리장치의 가격을 하락시킬 수 있고, 보다 효율적인 테스트 및 디버깅을 할 수 있는 프로그램 테스트 및 디버깅이 용이한 중앙처리장치 및 프로그램 테스트 및 디버깅 방법을 제공하는 데 있다.

【발명의 구성 및 작용】

<25> 상기의 목적을 달성하기 위하여 본 발명의 프로그램 테스트 및 디버깅이 용이한 중앙처리장치는 호스트 컴퓨터와 데이터 통신을 위한 데이터 통신부; 중앙처리장치의 동작모드가 일반 동작 상태를 나타내는 일반 동작 모드인지 디버깅 상태를 나타내는 디버

깅 모드인지를 나타내는 1 이상의 플래그들을 구비한 상태 레지스터; 메모리상의 디버깅 스택 영역을 지정하는 디버깅 스택 포인터 레지스터; 브레이크 레지스터에 저장된 값과 중지데이터를 비교하여 동일하면 중앙처리장치를 디버깅 모드로 진입시키는 비교부를 구비하여 브레이크 레지스터에 저장된 값과 중지 데이터가 일치하면 중앙처리장치를 디버깅 모드로 전환시키고 1이상의 플래그들이 디버깅 모드를 나타내는 값을 가지며 디버깅 스택 포인터 레지스터를 디버깅 프로그램의 데이터를 저장하는 스택 포인터로 사용하며 데이터 통신부를 이용하여 호스트 컴퓨터로부터의 명령에 따라 디버깅을 수행하는 것을 특징으로 한다.

<26> 비교부의 중지데이터(BD)는 프로그램 카운터에 저장된 프로그램 번지, 데이터가 저장된 메모리 어드레스, 입출력되는 데이터, 입출력되는 어드레스일 수 있고, 마스크 레지스터(MR)를 더 구비하여 프로그램 카운터에 저장된 프로그램 번지와 마스크 레지스터(MR)에 저장된 값을 연산한 결과, 데이터가 저장된 메모리 어드레스와 마스크 레지스터(MR)에 저장된 값을 연산한 결과, 입출력되는 어드레스와 마스크 레지스터(MR)에 저장된 값을 연산한 결과, 입출력되는 데이터와 마스크 레지스터(MR)에 저장된 값을 연산한 결과인 것을 특징으로 한다.

<27> 이하, 첨부된 도면을 참조하여 본 발명의 프로그램 테스트 및 디버깅이 용이한 중앙처리장치와 프로그램 테스트 및 디버깅 방법을 상세히 설명하고자 한다.

<28> 도 3은 본 발명의 프로그램 테스트 및 디버깅이 용이한 중앙처리장치를 도시한 블록다이어그램이다.

<29> 도 3의 본 발명의 프로그램 테스트 및 디버깅이 용이한 중앙처리장치는 데이터 및 어드레스 연산에 필요한 데이터를 일시적으로 저장하는 범용 레지스터 파일(100), 프로그램이 저장된 메모리의 주소를 기억하는 프로그램 카운터(PC)와 중앙처리장치의 일반 동작 상태를 나타내는 일반 동작 모드 및 프로그램 테스트 및 디버깅을 하기 위한 상태를 나타내는 디버깅 모드를 나타내는 디버깅 모드 플래그를 가진 상태 레지스터(SR)와 브레이크 레지스터(BR), 마스크 레지스터 및 디버깅 스택 포인터(ISP)로 구성된 특수 레지스터 파일(200), 범용 레지스터 파일(100) 및 특수 레지스터 파일(200)을 연결하는 내부 버스(IB), 내부 버스(IB)에 연결되어 중앙처리장치 내외부에 필요한 각종 제어신호(CONT)를 출력하는 제어 유닛(300), 메모리 어드레스 레지스터(400), 메모리 데이터 레지스터(500), 호스트 컴퓨터와 데이터 통신을 위한 데이터 통신부(600) 및 브레이크 레지스터(BR)에 저장된 값(IBR)과 중지데이터(BD)를 비교하여 동일하면 중앙처리장치를 디버깅 모드로 진입시키는 비교부(700)로 구성되어 리셋 신호(RESET)에 의해 리셋 상태로 전환되어 상기의 중앙처리장치를 초기화하고 리셋상태에서 벗어날 때 제어 유닛(300)로 입력되는 디버깅 모드 진입 결정 신호(OSI)가 활성화되면 도 4에 도시된 바와 같이 상태 레지스터(SR)의 디버깅 모드 플래그(OSIM)를 활성화시켜, 중앙처리장치를 응용 프로그램의 테스트 및 디버깅을 수행하기 위한 초기 환경으로 설정해 주고, 데이터 통신부(600)를 통해 호스트 컴퓨터로부터 중앙처리장치로 다운로드 된 테스트 및 디버깅을 하기 위한 응용 프로그램을 실행한 후, 비교부(700)에 의해 브레이크 레지스터(BR)에 저장된 값(IBR)과 중지데이터(BD)가 동일하면 중앙처리장치를 디버깅 모드로 진입시킨 후 데이터 통신부(600)를 통해 호스트 컴퓨터로부터의 명령에 따라 응용 프로그램의 테스트 및 디버깅을 수행한다.

<30> 응용 프로그램은 디버깅 모드 진입 결정 신호(OSI)가 활성화되어 테스트 및 디버깅을 수행하기 위한 초기 환경으로 중앙처리장치를 설정한 후 호스트 컴퓨터로부터 중앙처리장치로 다운로드 받거나, 중앙처리장치를 테스트 및 디버깅을 수행하기 위한 초기 환경으로 설정한 후 비교부(700)에 의해 브레이크 레지스터(BR)에 저장된 값(IBR)과 중지 데이터(BD)가 동일하면 중앙처리장치를 프로그램 테스트 및 디버깅 상태로 진입시킨 후 호스트 컴퓨터로부터 중앙처리장치로 다운로드 받을 수 있다.

<31> 비교부(700)에 의해 브레이크 레지스터(BR)에 저장된 값(IBR)과 중지데이터(BD)가 동일하면 중앙처리장치는 응용 프로그램의 실행 중인 동작을 중지 또는 완료하고, 프로그램 카운터(PC)에 저장된 값과 상태 레지스터(SR)에 저장된 데이터를 임시 저장 레지스터(TR)나 디버깅 스택 포인터(ISP)가 지정하는 스택에 저장하고, 프로그램 카운터(PC)에는 응용 프로그램을 테스트 및 디버깅하기 위한 디버깅 서비스 프로그램을 수행하기 위한 시작 주소를 저장하고, 중앙처리장치를 응용 프로그램을 테스트 및 디버깅하기 위한 상태에 있도록 상태 레지스터(SR)의 데이터를 변경한다.

<32> 본 발명의 프로그램 테스트 및 디버깅이 용이한 중앙처리장치는 사용자가 호스트 컴퓨터에서 명령을 내려, 응용 프로그램을 수행중인 중앙처리장치의 모드를 강제로 디버깅 모드로 전환하기 위해, 미리 설정된 기준 데이터를 저장하는 기준 데이터 저장부(900)와 기준 데이터 저장부(900)에 저장된 기준데이터와 데이터 통신부(600)를 통해 입력된 디버깅 데이터(DEB)를 비교하는 기준 데이터 비교부(800)를 더 구비하여, 비교 결과 두 값이 동일하면 중앙처리장치를 프로그램 테스트 및 디버깅 상태로 진입시킨다.

<33> 본 발명의 프로그램 테스트 및 디버깅이 용이한 중앙처리장치는 사용자가 호스트 컴퓨터에서 명령을 내려 중앙처리장치를 초기화시키기 위하여, 미리 설정된 리셋 데이터

를 저장하는 리셋 데이터 저장부(1100)와 리셋 데이터 저장부(1100)에 저장된 리셋데이터와 데이터 통신부(600)을 통해 입력된 디버깅 데이터(DEB)를 비교하는 리셋 데이터 비교부(1000)를 더 구비하여, 비교 결과 두 값이 동일하면 중앙처리장치를 초기화시킨다.

<34> 본 발명의 프로그램 테스트 및 디버깅이 용이한 중앙처리장치는 빠른 디버깅을 위하여 임시 저장 레지스터(1200)를 더 구비하여 디버깅 모드로 전환한 뒤 디버깅 프로그램을 수행하기 전에 프로그램 카운터(PC)와 상태 레지스터(SR)를 임시 저장 레지스터(1200)에 저장하고 디버깅 프로그램을 수행 할 수도 있다.

<35> 도 5a에 도시된 바와 같이 중앙처리장치의 초기화 및 일반 동작을 위한 응용 프로그램(PROG1; PROG2, PROG4), 디버깅 모드 진입 결정 신호(OSI)의 활성화에 따라 중앙처리장치를 응용 프로그램의 테스트 및 디버깅을 수행하기 위한 초기 환경으로 설정해 주는 디버깅 모드 초기화 프로그램(PROG3, PROG6)과 비교부(700)에 의해 중앙처리장치를 프로그램 테스트 및 디버깅 상태로 진입시켜 응용 프로그램을 테스트 및 디버깅을 위한 디버깅 서비스 프로그램(PROG5, PROG7)은 동일한 메모리(800)에 저장되거나 도 5b에 도시된 바와 같이 다른 독립된 메모리(900)에 저장될 수 있다.

<36> 또는 도 5c에 도시된 바와 같이 디버깅 모드 초기화 프로그램(PROG3, PROG6) 및 디버깅 서비스 프로그램(PROG5, PROG7)은 중앙처리장치를 초기화 및 일반 동작을 위한 응용 프로그램(PROG1, PROG2, PROG4)이 저장된 메모리(800) 및 다른 독립된 메모리(900)에 모두 존재할 수 있다.

<37> 도 6에 도시된 바와 같이 본 발명의 프로그램 테스트 및 디버깅 방법은 중앙처리장치를 초기화시키는 리셋 신호(RESET)의 활성화 여부를 판단하는 리셋 판단 단계(S10), 리셋 판단 단계(S10)에서 리셋 신호(RESET)가 활성화되면 중앙처리장치를 응용 프로그램

의 테스트 및 디버깅을 수행하기 위한 초기 환경으로 설정해 주기 위한 디버깅 모드 진입 결정 신호(OSI)의 활성화 여부를 판단하는 디버깅 진입 판단 단계(S20), 디버깅 모드 진입 판단 단계(S20)에서 디버깅 모드 진입 결정 신호(OSI)가 활성화되면 중앙처리장치를 응용 프로그램의 테스트 및 디버깅을 수행하기 위한 초기 환경으로 설정해 주는 디버깅 모드 초기화 단계(S30), 디버깅 모드 초기화 단계 (S30) 후 응용 프로그램을 실행하는 실행 단계(S40), 실행 단계(S40)에서 실행 중인 응용 프로그램을 테스트 및 디버깅 상태로 진입시켜 호스트 컴퓨터로부터 입력된 명령어를 수행하여 응용 프로그램의 테스트 및 디버깅을 수행하는 디버깅 단계(S50)로 구성된다.

<38> 디버깅 단계(S50)는 브레이크 레지스터(BR)에 저장된 값과 중지데이터(BD)가 동일한 지를 비교하는 비교단계(S51), 비교단계(S51)에서 브레이크 레지스터(BR)에 저장된 값과 중지데이터가 동일하면 중앙처리장치가 실행 중인 동작을 중지 또는 완료하고, 프로그램 카운터에 저장된 값과 상태 레지스터에 저장된 데이터를 별도의 임시 저장 레지스터(TR)나 디버깅 스택 포인터(ISP)가 지정하는 스택에 저장하는 저장 단계(S52), 프로그램 카운터에는 응용 프로그램을 테스트 및 디버깅하기 위한 디버깅용 프로그램을 수행하기 위한 시작 주소를 저장하고, 중앙처리장치를 응용 프로그램을 테스트 및 디버깅하기 위한 상태에 있도록 상태 레지스터의 데이터를 변경하여 디버깅 서비스 프로그램을 수행하는 디버깅 서비스 프로그램 실행단계(S53)로 구성된다.

<39> 리셋 판단 단계(S10)에서 리셋 신호(RESET)는 호스트 컴퓨터로부터 데이터 통신부(600)로 입력되는 디버깅 데이터(DEB)가 리셋 데이터 저장부(1100)에 저장된 리셋 데이터와 리셋 데이터 비교부(100)에 의해 비교되어 두 값이 같을 때 활성화될 수 있다.

<40> 비교부의 중지데이터(BD)는 프로그램 카운터에 저장된 프로그램 번지, 데이

터가 저장된 메모리 어드레스, 입출력되는 데이터, 입출력되는 어드레스일 수 있고, 마스크 레지스터(MR)를 더 구비하여 프로그램 카운터에 저장된 프로그램 번지와 마스크 레지스터(MR)에 저장된 값을 연산한 결과, 데이터가 저장된 메모리 어드레스와 마스크 레지스터(MR)에 저장된 값을 연산한 결과, 입출력되는 어드레스와 마스크 레지스터(MR)에 저장된 값을 연산한 결과, 입출력되는 데이터와 마스크 레지스터(MR)에 저장된 값을 연산한 결과인 것을 특징으로 한다. 디버깅 모드 초기화 단계(S30)에서의 중앙처리장치를 응용 프로그램의 테스트 및 디버깅을 수행하기 위한 초기 환경으로 설정해 주는 디버깅 모드 초기화 프로그램과 디버깅 단계(S50)에서의 응용 프로그램의 테스트 및 디버깅을 수행하기 위한 디버깅 서비스 프로그램은 중앙처리장치를 초기화 및 일반 동작을 위한 서비스 프로그램이 저장된 동일한 메모리에 저장될 수도 있고 다른 메모리에 저장될 수도 있다.

<41> 도 7에 도시된 바와 같이 본 발명의 다른 실시 예인 프로그램 테스트 및 디버깅 방법은 중앙처리장치를 초기화시키는 리셋 신호의 활성화 여부를 판단하는 리셋 판단 단계(S10a), 리셋 판단 단계(S10a)에서 리셋 신호(RESET)가 활성화되면 중앙처리장치를 응용 프로그램의 테스트 및 디버깅을 수행하기 위한 초기 환경으로 설정해 주기 위한 디버깅 모드 진입 결정 신호(OSI)의 활성화 여부를 판단하는 디버깅 모드 진입 판단 단계(S20a), 중앙처리장치를 응용 프로그램의 테스트 및 디버깅을 수행하기 위한 초기 환경으로 설정해 주는 디버깅 모드 초기화 프로그램(PROG3, PROG6)과 응용 프로그램을 테스트 및 디버깅을 위한 디버깅 서비스 프로그램(PROG5, PROG7)이 중앙처리장치의 초기화 및 일반 동작을 위한 서비스 프로그램이 저장되어 있는 제1메모리(800)에 저장되고 동시에 제1메모리(800)와 독립된 제2메모리(900)에 저장되어 있는 경우 제1메모리(800)에 저

장된 디버깅 모드 초기화 프로그램(PROG3) 및 디버깅 서비스 프로그램(PROG5)을 사용할 것인지 또는 제2메모리(900)에 저장된 디버깅 모드 초기화 프로그램(PROG6) 및 디버깅 서비스 프로그램(PROG7)을 사용할 것인지를 구분하는 메모리 구분 신호(OSIROM)의 활성화 여부를 판단하는 메모리 사용 판단 단계(S60), 디버깅 모드 진입 결정 신호(OSI)가 활성화되고 메모리 구분 신호(OSIROM)가 비활성화되면 제1메모리(800)에 저장된 디버깅 모드 초기화 프로그램(PROG3)에 따라 중앙처리장치를 응용 프로그램의 테스트 및 디버깅을 수행하기 위한 초기 환경으로 설정해 주고, 디버깅 모드 진입 결정 신호(OSI)가 활성화되고 메모리 구분 신호(OSIROM)가 활성화되면 제2메모리(900)에 저장된 디버깅 모드 초기화 프로그램(PROG6)에 따라 중앙처리장치를 응용 프로그램의 테스트 및 디버깅을 수행하기 위한 초기 환경으로 설정해 주는 디버깅 모드 초기화 단계(S30a), 디버깅 모드 초기화 단계(S30a) 후 응용 프로그램을 실행하는 실행 단계(S40a), 메모리 구분 신호(OSIROM)가 비활성화되면 제1메모리(800)에 저장된 디버깅 서비스 프로그램(PROG5)에 의해 실행 단계(S40a)에서 실행 중인 응용 프로그램을 테스트 및 디버깅 상태로 진입시키고, 메모리 구분 신호(OSIROM)가 활성화되면 제2메모리(900)에 저장된 디버깅 서비스 프로그램(PROG7)에 의해 실행 단계(S40a)에서 실행 중인 응용 프로그램을 테스트 및 디버깅 상태로 진입시켜 호스트 컴퓨터로부터 입력된 명령어를 수행하여 응용 프로그램의 테스트 및 디버깅을 수행하는 디버깅 단계(S50a)로 구성된다.

<42> 디버깅 단계(S50a)는 도 6과 동일하게 비교단계(S51a), 저장 단계(S52a) 및 디버깅 서비스 프로그램 실행 단계(S53a)로 구성된다.

<43> 상기의 구성에 따른 본 발명인 프로그램의 테스트 및 디버깅이 용이한 중앙처리장

치의 동작은 다음과 같다.

- <44> 도 5a에 도시된 바와 같이 중앙처리장치를 초기화 및 일반 동작을 위한 응용 프로그램(PROG1, PROG2, PROG4), 디버깅 모드 초기화 프로그램(PROG3) 및 디버깅 서비스 프로그램(PROG5)이 모두 동일한 제1메모리(800)에 저장되어 있는 경우 본 발명의 동작은 다음과 같다.
- <45> 중앙처리장치에 전원이 공급되면 리셋 신호(RESET)가 활성화된다. 리셋 신호(RESET)가 활성화되면 제어 유닛(300)은 디버깅 모드 진입 결정 신호(OSI)의 활성화 여부를 판단한다. 디버깅 모드 진입 결정 신호(OSI)가 비활성화되어 있으면 도 4의 상태 레지스터(SR)의 디버깅 모드 플래그(OSIM)를 비활성화시킨다. 제어 유닛(300)에 입력되는 활성화된 리셋 신호(RESET)에 의하여 제어 유닛(300)은 최초로 수행할 프로그램이 저장된 메모리(800) 주소가 저장되어 있는 벡터 테이블(810)에서 도시된 바와 같이 리셋 벡터를 출력하여 특수 레지스터 파일(200)의 프로그램 카운터(PC)에 저장한다. 프로그램 카운터(PC)에 저장된 리셋 벡터는 메모리 어드레스 레지스터(400)로 이동되어 어드레스 버스(ADDB)로 출력하여 메모리(800)에 저장된 응용 프로그램(PROG1)을 수행한다. 이때 제어 유닛(300)은 메모리 읽기 신호를 제어버스(CONT)로 출력하여 응용 프로그램(PROG1)이 저장된 메모리(800) 내용을 읽어 들이게 된다. 메모리에서 출력된 데이터는 데이터 버스(DATAB) 및 메모리 데이터 레지스터(500)를 통하여 내부버스(IB)를 경유하여 제어 유닛(300)로 들어오게 된다. 입력된 명령어는 해석되어 제어 유닛(300)으로부터 필요한 제어신호들(CONT)이 출력되어 해당 명령어를 수행한다.
- <46> 리셋 벡터에 의해 디버깅하기 위한 응용 프로그램(PROG1)을 수행하기 위해 중앙처리장치가 초기화되면, 리셋 신호(RESET)는 비활성화되고 중앙처리장치는 일반적인 동작

을 수행한다.

<47> 만일 리셋 신호(RESET)가 활성화된 후 제어 유닛(300)로 입력되는 디버깅 모드 진입 결정 신호(OSI)가 활성화되면, 상태 레지스터(SR)의 디버깅 모드 플래그(OSIM)를 활성화시켜 중앙처리장치를 응용 프로그램의 테스트 및 디버깅을 수행하기 위한 초기 환경으로 설정해 주기 위한 디버깅 상태로 만들어 준다.

<48> 활성화된 디버깅 모드 진입 결정 신호(OSI)에 의하여 제어 유닛(300)는 메모리(800)에 저장된 디버깅 모드 초기화 프로그램(PROG3)의 시작 주소인 디버깅 모드 초기화 벡터를 출력하고, 디버깅 모드 초기화 벡터를 프로그램 카운터(PC)에 저장한다. 프로그램 카운터(PC)에 저장된 디버깅 모드 초기화 벡터에 의해 디버깅 모드 초기화 프로그램(PROG3)을 수행한 후, 중앙처리장치는 응용 프로그램의 테스트 및 디버깅을 수행하기 위한 초기 환경으로 설정된다.

<49> 호스트 컴퓨터에서의 명령에 따라 디버깅하기 위한 응용 프로그램을 실행시키기 위해서 중앙처리장치는 상태 레지스터(SR)의 디버깅 모드 플래그(OSIM)를 비활성화시켜 디버깅을 하기 위한 응용 프로그램을 실행한다. 응용 프로그램의 실행 중 비교부(700)에 의해 특수 레지스터 파일(200)의 브레이크 레지스터(BR)에 저장된 값(IBR)과 중지데이터(BD)가 동일하면, 중앙처리장치는 응용 프로그램의 실행 중인 동작을 중지 또는 완료하고, 프로그램 카운터(PC)에 저장된 값과 상태 레지스터(SR)에 저장된 데이터를 임시 저장 레지스터(TR)에 저장하고, 프로그램 카운터(PC)에는 응용 프로그램을 테스트 및 디버깅하기 위한 디버깅 서비스 프로그램(PROG5)을 수행하기 위한 시작 주소인 디버깅 서비스 벡터를 저장하고, 응용 프로그램을 테스트 및 디버깅하기 위한 상태에 있도록 중앙처리장치의 상태 레지스터(SR) 값을 변경한다. 즉 상태 레지스터(SR)의 디버깅 모드 플래

그(OSIM)를 활성화시킨다.

<50> 비교부(700)에 의해 브레이크 레지스터(BR)에 저장된 값(IBR)과 중지데이터(BD)가 동일하면 중앙처리장치는 응용 프로그램의 실행 중인 동작을 중지 또는 완료하고, 프로그램 카운터(PC)에 저장된 값과 상태 레지스터(SR)에 저장된 데이터를 별도의 임시 저장 레지스터(TR)에 저장하지 않고 디버깅 스택 포인터(ISP)가 지정하는 스택에 저장할 수도 있다.

<51> 중앙처리장치는 프로그램 테스트 및 디버깅 상태로 진입한 후 메모리(800)에 저장된 디버깅 서비스 프로그램(PROG5)에 따라 호스트 컴퓨터로부터의 명령, 예를 들어 사용자가 작성한 응용 프로그램의 다운로드, 다운로드된 응용 프로그램의 실행, 브레이크 레지스터(BR)의 데이터 변경, 범용 레지스터 파일(100) 및 특수 레지스터 파일(200)의 레지스터들에 저장된 데이터의 판독(READ) 또는 기록(WRITE) 및 메모리에 저장된 데이터들의 판독 또는 기록 등의 명령을 수행하여 응용 프로그램이 올바르게 작성되었는지를 판단하고, 잘못 작성되었으면 이를 수정하여 바로 잡는다.

<52> 상기의 명령들을 수행하기 위하여 데이터 통신부(600)를 통해 중앙처리장치와 호스트 컴퓨터간의 처리 순서는 다음과 같다.

<53> 호스트 컴퓨터로부터의 다운로드 명령에 의해 사용자가 작성한 프로그램을 데이터 통신부(600)를 통해 전송받아 중앙처리장치의 메모리에 저장한 후 중앙처리장치는 성공적으로 수신 받았다는 신호를 호스트 컴퓨터로 송신한다. 만약 응용 프로그램의 전송 중에러가 발생했다면 중앙처리장치는 호스트 컴퓨터로 응용 프로그램의 재전송을 요구한다. 응용 프로그램을 중앙처리장치에 성공적으로 다운로드 한 후 디버깅을 위하여 호스트 컴퓨터는 중앙처리장치의 브레이크 레지스터(BR)의 데이터 값을 설정하기 위한 브레

이크 레지스터(BR) 데이터 변경 명령을 내린다. 브레이크 레지스터(BR) 데이터 변경 명령 후 중앙처리장치는 호스트 컴퓨터로 성공적으로 수신 받았다는 신호를 송신한다. 호스트 컴퓨터는 중앙처리장치의 메모리에 다운로드된 응용 프로그램을 실행시키기 위하여 응용 프로그램의 실행 명령을 중앙처리장치로 송신하고, 중앙처리장치는 이에 응답하고 응용 프로그램을 실행한다. 응용 프로그램의 실행 중 비교부(700)에 의해 브레이크 레지스터(BR)의 데이터 값과 중지데이터인 프로그램 카운터(PC)의 데이터 값이 동일하면, 응용 프로그램의 실행 중인 동작을 중지 또는 완료하고 프로그램 카운터(PC)에 저장된 값과 상태 레지스터(SR)에 저장된 데이터 및 범용 레지스터의 내용들을 임시 저장 레지스터(TR) 또는 디버깅 스택 포인터(ISP)가 지정하는 스택에 저장한다. 프로그램 카운터(PC)에는 응용 프로그램을 테스트 및 디버깅하기 위한 디버깅 서비스 프로그램(PROG5)을 수행하기 위한 시작 주소인 디버깅 서비스 벡터를 저장하고, 중앙처리장치를 응용 프로그램을 테스트 및 디버깅하기 위한 상태에 있도록 상태 레지스터(SR)의 데이터를 변경한다. 즉 상태 레지스터(SR)의 디버깅 모드 플래그(OSIM)를 활성화시킨다. 디버깅 서비스 프로그램(PROG5)에 의해 중앙처리장치는 응용 프로그램의 테스트 및 디버깅을 수행하기 위한 상태로 진입된다. 중앙처리장치가 테스트 및 디버깅을 수행하기 위한 디버깅 모드로 진입한 후 호스트 컴퓨터로부터 레지스터 판독 명령을 수신 받으면, 임시 저장 레지스터(TR) 또는 디버깅 스택 포인터(ISP)가 지정하는 스택에 저장된 레지스터의 데이터를 호스트 컴퓨터로 전송하고, 사용자는 전송된 데이터를 분석하여 올바른 값이 출력되었는지를 판단하고, 분석 결과 올바른 데이터가 아니면 응용 프로그램이 올바르게 작성되지 않은 것이므로 응용 프로그램을 수정한다.

<54> 상기와 같은 방법을 반복하여 응용 프로그램을 용이하게 디버깅할 수 있다.

- <55> 리셋 신호(RESET)는 중앙처리장치에 최초 전압 인가시 활성화되거나 리셋 데이터 비교부(1000)에 의해서 데이터 통신부(600)를 통해 호스트 컴퓨터로부터 입력되는 디버깅 데이터(DEB)와 리셋 데이터 저장부(1100)에 저장된 리셋 데이터가 비교되어 동일할 경우 활성화될 수 있다.
- <56> 응용 프로그램이 일정 시간 이상 계속해서 실행되는 경우, 즉 응용 프로그램이 무한 루프에 있는 경우 데이터 통신부(600)를 통해 기준 데이터 저장부(900)에 저장된 기준데이터와 동일한 디버깅 데이터를 송신하면, 기준 데이터 비교부(800)에 의해 기준 데이터 저장부에 저장된 기준 데이터와 데이터 통신부(600)를 통해 받은 디버깅 데이터를 비교하여 같으면 강제적으로 중앙처리장치를 프로그램 테스트 및 디버깅 상태인 디버깅 모드로 진입시켜 무한 루프에서 벗어나게 할 수 있다.
- <57> 도 5c에 도시된 바와 같이 디버깅 모드 초기화 프로그램(PROG3, PROG6) 및 디버깅 서비스 프로그램(PROG5, PROG7)은 중앙처리장치를 초기화 및 동작시키기 위한 서비스 프로그램이 저장된 제1메모리(800)와 별도의 제2메모리(900)에 동시에 저장될 수 있다. 이를 위해 제어 유닛(300)의 입력신호로 메모리 구분 신호(OSIROM)가 필요하다. 즉 리셋 신호(RESET)가 활성화되고, 디버깅 모드 진입 결정 신호(OSI)가 활성화되고, 메모리 구분 신호(OSIROM)가 활성화되면 응용 프로그램의 테스트 및 디버깅을 위해 중앙처리장치를 초기화 시켜주기 위해 프로그램 카운터(PC)에는 제2메모리(900)에 저장된 디버깅 모드 초기화 프로그램(PROG6)의 시작 주소인 디버깅 모드 초기화 벡터2가 저장되어야 하고, 비교부(700)에 의해 중앙처리장치를 프로그램 테스트 및 디버깅 상태로 진입시키기 위해 프로그램 카운터(PC)에는 제2메모리(900)에 저장된 디버깅 서비스 프로그램(PROG7)의 시작 주소인 디버깅 서비스 벡터2가 저장되어야 한다.

- <58> 만약에 메모리 구분 신호(OSIROM)가 비활성화되면, 디버깅 모드 초기화 벡터1이 디버깅 모드 초기화 프로그램(PROG3)을 수행하기 위한 시작 주소이고 디버깅 서비스 벡터1이 디버깅 서비스 프로그램(PROG5)을 수행하기 위한 시작 주소이다.
- <59> 상기와 같이 메모리를 분리함으로써 사용자는 내장되어 있는 디버깅 모드 초기화 프로그램(PROG6) 및 디버깅 서비스 프로그램(PROG7)을 사용하지 않고, 사용자가 작성한 디버깅 모드 초기화 프로그램(PROG3) 및 디버깅 서비스 프로그램(PROG5)을 이용하여 응용 프로그램을 테스트 및 디버깅할 수 있다.
- <60> 도 6에 도시된 본 발명의 프로그램 테스트 및 디버깅 방법의 동작은 다음과 같다.
- <61> 리셋 판단 단계(S10)는 중앙처리장치를 초기화시키는 리셋 신호(RESET)의 활성화 상태를 판단한다. 디버깅 판단 단계(S20)는 리셋 신호(RESET)가 활성화되었을 때 중앙처리장치를 응용 프로그램의 테스트 및 디버깅을 수행하기 위한 초기 환경으로 설정해 주기 위한 디버깅 모드 진입 결정 신호(OSI)의 활성화 여부를 판단한다. 디버깅 모드 셋팅 단계(S30)는 디버깅 모드 진입 결정 신호(OSI)가 활성화되었을 때 중앙처리장치를 응용 프로그램의 테스트 및 디버깅을 수행하기 위한 초기 환경으로 설정해 준다. 실행 단계(S40)는 호스트 컴퓨터로부터 다운로드 받은 응용 프로그램을 실행한다. 디버깅 단계(S50)는 중앙처리장치를 디버깅 모드로 전환시키고 호스트 컴퓨터로부터 입력된 명령어를 수행하여 응용 프로그램의 테스트 및 디버깅을 수행한다.
- <62> 디버깅 단계(S50)는 비교단계(S51), 저장 단계(S52) 및 디버깅 서비스 프로그램 실행단계(S53)로 구성된다.
- <63> 비교단계(S51)는 브레이크 레지스터(BR)에 저장된 값과 프로그램 카운터에 저장된

데이터, 데이터가 저장된 메모리 어드레스 또는 입출력장치의 어드레스인 중지데이터(BD)가 동일한 지를 비교한다. 저장 단계(S52)는 브레이크 레지스터(BR)에 저장된 값과 중지데이터(BD)가 동일할 경우 중앙처리장치는 실행 중인 동작을 중지하고, 프로그램 카운터에 저장된 값과 상태 레지스터에 저장된 데이터를 별도의 임시 저장 레지스터(TR)나 디버깅 스택 포인터(ISP)가 지정하는 스택에 저장한다. 디버깅 서비스 프로그램 실행단계(S53)는 프로그램 카운터(PC)에 응용 프로그램을 테스트 및 디버깅하기 위한 디버깅 서비스 프로그램을 수행하기 위한 시작 주소를 저장하고, 중앙처리장치를 응용 프로그램을 테스트 및 디버깅하기 위한 상태에 있도록 상태 레지스터의 데이터를 변경하여 디버깅용 프로그램을 수행한다.

<64> 리셋 신호(RESET)는 중앙처리장치에 최초 전압 인가시 활성화되거나 리셋 데이터 비교부(1000)에 의해서 데이터 통신부(600)를 통해 호스트 컴퓨터로부터 입력되는 디버깅 데이터(DEB)와 리셋 데이터 저장부(1100)에 저장된 리셋 데이터가 비교되어 동일할 경우 활성화될 수 있다.

<65> 도 5a 및 도 5b에 도시된 바와 같이 디버깅 모드 초기화 단계(S30)에서의 중앙처리장치를 응용 프로그램의 테스트 및 디버깅을 수행하기 위한 초기 환경으로 설정해 주는 디버깅 모드 초기화 프로그램(PROG3, PROG6) 및 디버깅 단계(S50)에서의 응용 프로그램의 테스트 및 디버깅을 수행하기 위한 디버깅 서비스 프로그램(PROG5, PROG7)은 중앙처리장치를 초기화 및 일반 동작을 위한 응용 프로그램(PROG1, PROG2, PROG4)이 저장된 제1메모리(800)에 저장되거나 제2메모리(900)에 저장될 수 있다.

<66> 도 7에 도시된 바와 같이 본 발명의 다른 실시 예인 프로그램 테스트 및 디버깅 방법의 동작은 도 6에 도시된 프로그램 테스트 및 디버깅 방법의 동작과 동일

하나 도 7의 본 발명의 다른 실시 예인 프로그램 테스트 및 디버깅 방법은 도 5c에 도시된 바와 같이 디버깅 모드 초기화 프로그램(PROG3, PROG6) 및 디버깅 서비스 프로그램(PROG5, PROG7)은 제1메모리(800)와 제2메모리(900)에 동시에 저장되어 있으므로 디버깅 모드 초기화 단계와 디버깅 단계시 어느 메모리에 저장된 디버깅 서비스 프로그램과 디버깅 서비스 프로그램을 사용할 것인지를 판단하는 메모리 사용 판단 단계(S60)가 더 필요하다.

<67> 리셋 판단 단계(S10a), 디버깅 판단 단계(S20a) 후의 메모리 사용 판단 단계(S60)는 제1메모리(800)에 저장된 디버깅 모드 초기화 프로그램(PROG3) 및 디버깅 서비스 프로그램(PROG5)을 사용할 것인지 또는 제2메모리(900)에 저장된 디버깅 모드 초기화 프로그램(PROG6) 및 디버깅 서비스 프로그램(PROG7)을 사용할 것인지를 구분하는 메모리 구분 신호(OSIROM)의 활성화 여부를 판단한다. 디버깅 셋팅 단계(S30a)는 디버깅 모드 진입 결정 신호(OSI)가 활성화되고 메모리 구분 신호(OSIROM)가 비활성화되면, 제1메모리(800)에 저장된 디버깅 모드 초기화 프로그램(PROG3)에 따라 중앙처리장치를 응용 프로그램의 테스트 및 디버깅을 수행하기 위한 초기 환경으로 설정해 준다. 하지만 디버깅 모드 진입 결정 신호(OSI)가 활성화되고 메모리 구분 신호(OSIROM)가 활성화되면, 제2메모리(900)에 저장된 디버깅 모드 초기화 프로그램(PROG6)에 따라 중앙처리장치를 응용 프로그램의 테스트 및 디버깅을 수행하기 위한 초기 환경으로 설정해 준다. 실행 단계(S40a)는 응용 프로그램을 실행한다. 디버깅 단계(S50a)는 메모리 구분 신호(OSIROM)가 비활성화되면 제1메모리(800)에 저장된 디버깅 서비스 프로그램(PROG5)에 의해 실행 단계(S40a)

에서 실행 중인 응용 프로그램을 중지 또는 완료한 후 중앙처리장치를 테스트 및 디버깅 상태로 진입시키고, 메모리 구분 신호(OSIROM)가 활성화되면 제2메모리(900)에 저장된 디버깅 서비스 프로그램(PROG7)에 의해 실행 단계(S40a)에서 실행 중인 응용 프로그램을 중지 또는 완료한 후 중앙처리장치를 테스트 및 디버깅 상태로 진입시켜 호스트 컴퓨터로부터 입력된 명령을 수행하여 응용 프로그램의 테스트 및 디버깅을 수행한다.

<68> 디버깅 단계(S50a)는 도 6의 경우와 동일하게 비교단계(S51a), 저장 단계(S52a) 및 디버깅 서비스 프로그램 실행단계(S53a)로 구성된다.

【발명의 효과】

<69> 본 발명의 프로그램 테스트 및 디버깅이 용이한 중앙처리장치는 사용자 모드를 갖는 중앙처리장치의 상태에 디버깅 모드를 새로이 추가하고, 디버깅 모드에서 호스트 컴퓨터로부터 다운로드 된 테스트 및 디버깅을 하기 위한 응용 프로그램을 용이하게 테스트 및 디버깅할 수 있게 해준다. 여기서 중앙처리장치는 보다 단순한 하드웨어로 구성되어 중앙처리장치의 가격을 하락시킬 수 있고 보다 효율적인 테스트 및 디버깅을 할 수 있다. 또한 응용 프로그램의 테스트 및 디버깅을 수행하기 위한 초기 환경으로 설정해주는 디버깅 모드 초기화 프로그램 및 중앙처리장치를 프로그램 테스트 및 디버깅 상태로 진입시켜 응용 프로그램을 테스트 및 디버깅을 위한 디버깅 서비스 프로그램은 중앙처리장치를 초기화 및 동작시키기 위한 사용자 응용 프로그램이 저장된 메모리와 다른 독립된 메모리에 저장함으로써 사용자는 제공된 디버깅 모드 초기화 프로그램 및 디버깅 서비스 프로그램을 고려하지 않고 사용자에게 의해 작성된 디버깅 프로그램을 이용하여 응용 프로그램을 작성할 수 있다.

【특허청구범위】

【청구항 1】

데이터 및 어드레스 연산에 필요한 데이터를 일시적으로 저장하는 범용 레지스터 파일, 프로그램이 저장된 메모리의 주소를 기억하는 프로그램 카운터와 중앙처리장치의 상태를 나타내는 상태 레지스터 및 브레이크 레지스터로 구성된 특수 레지스터 파일, 상기의 범용 레지스터 파일 및 특수 레지스터 파일을 연결하는 내부 버스, 상기의 내부 버스에 연결되어 중앙처리장치 내외부에 필요한 각종 제어신호를 출력하는 제어 유니트로 구성된 중앙처리장치에 있어서,

호스트 컴퓨터와 데이터 통신을 위한 데이터 통신수단;

중앙처리장치의 동작모드가 일반 동작 상태를 나타내는 일반 동작 모드인지 디버깅 상태를 나타내는 디버깅 모드인지를 나타내는 1 이상의 플래그들을 구비한 상태 레지스터;

메모리상의 디버깅 스택 영역을 지정하는 디버깅 스택 포인터 레지스터; 및

상기의 브레이크 레지스터에 저장된 값(IBR)과 중지데이터를 비교하여 동일하면 상기의 중앙처리장치를 디버깅 모드로 진입시키는 비교수단을 구비하여 상기의 브레이크 레지스터에 저장된 값과 상기의 중지 데이터가 일치하면 상기의 중앙처리장치를 디버깅 모드로 전환시키고 상기의 1이상의 플래그들이 디버깅 모드를 나타내는 값을 가지고 상기의 디버깅 스택 포인터 레지스터를 디버깅 프로그램의 데이터를 저장하는 메모리를 지정하는 스택 포인터로 사용하며 상기의 프로그램 카운터에는 디버깅 프로그램을 수행하기 위한 시작 주소를 적재하고 상기의 데이터 통신수단을 이용하여 호스트 컴퓨터로부터

의 명령에 따라 디버깅을 수행하는 것을 특징으로 하는 프로그램 테스트 및 디버깅이 용이한 중앙처리장치.

【청구항 2】

제1항에 있어서, 상기의 프로그램 테스트 및 디버깅이 용이한 중앙처리장치는

설정된 리셋 데이터가 저장된 리셋 데이터 저장수단; 및

상기의 데이터 통신수단을 통해 입력받은 데이터와 상기의 리셋 데이터 저장수단에 저장된 설정된 리셋 데이터를 비교하여 상기의 데이터 통신수단을 통해 입력받은 데이터와 설정된 리셋 데이터와 같으면 상기의 중앙처리장치를 초기화하는 리셋 데이터 비교수단을 더 구비한 것을 특징으로 하는 프로그램 테스트 및 디버깅이 용이한 중앙처리장치.

【청구항 3】

제1항에 있어서, 상기의 중지데이터는 상기의 프로그램 카운터에 저장된 프로그램 주소인 것을 특징으로 하는 프로그램 테스트 및 디버깅이 용이한 중앙처리장치.

【청구항 4】

제1항에 있어서, 마스크 레지스터를 더 구비하여 상기의 중지데이터는 상기의 프로그램 카운터와 상기의 마스크 레지스터에 저장된 값을 연산한 결과인 것을 특징으로 하는 프로그램 테스트 및 디버깅이 용이한 중앙처리장치.

【청구항 5】

제1항에 있어서, 상기의 중지데이터는 데이터가 저장된 메모리 어드레스인 것을 특징으로 하는 프로그램 테스트 및 디버깅이 용이한 중앙처리장치.

【청구항 6】

제1항에 있어서, 마스크 레지스터를 더 구비하여 상기의 중지데이터는 데이터가 저장된 메모리 어드레스와 상기의 마스크 레지스터에 저장된 값을 연산한 결과인 것을 특징으로 하는 프로그램 테스트 및 디버깅이 용이한 중앙처리장치.

【청구항 7】

제1항에 있어서, 상기의 중지데이터는 상기의 중앙처리장치에서 입출력되는 데이터인 것을 특징으로 하는 프로그램 테스트 및 디버깅이 용이한 중앙처리장치.

【청구항 8】

제1항에 있어서, 마스크 레지스터를 더 구비하여 상기의 중지데이터는 상기의 중앙처리장치에서 입출력되는 데이터와 상기의 마스크 레지스터에 저장된 값을 연산한 결과인 것을 특징으로 하는 프로그램 테스트 및 디버깅이 용이한 중앙처리장치.

【청구항 9】

제1항에 있어서, 상기의 중지데이터는 상기의 중앙처리장치에서 입출력되는 어드레스인 것을 특징으로 하는 프로그램 테스트 및 디버깅이 용이한 중앙처리장치.

【청구항 10】

제1항에 있어서, 마스크 레지스터를 더 구비하여 상기의 중지데이터는 상기의 중앙처리장치에서 입출력되는 어드레스와 상기의 마스크 레지스터에 저장된 값을 연산한 결과인 것을 특징으로 하는 프로그램 테스트 및 디버깅이 용이한 중앙처리장치.

【청구항 11】

제1항에 있어서, 상기의 제어 유닛은 디버깅 메모리 선택신호를 수신하여 상기의

브레이크 레지스터에 저장된 값과 상기의 중지데이터가 동일할 때 상기의 디버깅 메모리 선택신호에 따라 상기의 프로그램 카운터에 서로 다른 디버깅 프로그램을 수행하기 위한 주소를 적재하는 것을 특징으로 하는 프로그램 테스트 및 디버깅이 용이한 중앙처리장치.

【청구항 12】

제1항에 있어서, 디버깅 모드로 전환되었을 때 수행되는 디버깅 프로그램에서 사용하는 데이터 값을 저장하는 데이터 저장 메모리를 더 구비하여 일반 프로그램에서 사용하는 데이터 저장 메모리를 분리한 것을 특징으로 하는 프로그램 테스트 및 디버깅이 용이한 중앙처리장치.

【청구항 13】

제1항에 있어서, 디버깅 모드로 전환되었을 때 수행되는 디버깅 프로그램을 저장하는 메모리를 더 구비하여 일반 프로그램이 저장된 메모리와 분리한 것을 특징으로 하는 프로그램 테스트 및 디버깅이 용이한 중앙처리장치.

【청구항 14】

제1항에 있어서, 상기의 데이터 통신 수단을 통해 상기의 호스트 컴퓨터로부터 테스트 및 디버깅을 하기 위한 디버깅 프로그램을 다운로드 받는 것을 특징으로 하는 프로그램 테스트 및 디버깅이 용이한 중앙처리장치.

【청구항 15】

제1항에 있어서, 상기의 데이터 통신 수단을 통해 상기의 호스트 컴퓨터로부터 테스트 및 디버깅 할 응용 프로그램을 다운로드 받아 메모리에 저장하고, 테스트 및 디버

강 할 응용 프로그램을 테스트 및 디버깅하는 것을 특징으로 하는 프로그램 테스트 및 디버깅이 용이한 중앙처리장치.

【청구항 16】

제1항에 있어서, 디버깅 모드로 전환되었을 때 상기의 프로그램 카운터에 저장된 값과 상기의 상태 레지스터에 저장된 데이터를 상기 디버깅 스택 포인터 레지스터가 지칭하는 메모리에 저장하는 것을 특징으로 하는 프로그램 테스트 및 디버깅이 용이한 중앙처리장치.

【청구항 17】

제1항에 있어서, 임시 저장 레지스터를 더 구비하여 디버깅 모드로 전환되었을 때 상기의 프로그램 카운터에 저장된 값과 상기의 상태 레지스터에 저장된 데이터를 상기 임시 저장 레지스터에 저장하는 것을 특징으로 하는 프로그램 테스트 및 디버깅이 용이한 중앙처리장치.

【청구항 18】

데이터 및 어드레스 연산에 필요한 데이터를 일시적으로 저장하는 범용 레지스터 파일, 프로그램이 저장된 메모리의 주소를 기억하는 프로그램 카운터와 중앙처리장치의 상태를 나타내는 상태 레지스터로 구성된 특수 레지스터 파일, 상기의 범용 레지스터 파일 및 특수 레지스터 파일을 연결하는 내부 버스, 상기의 내부 버스에 연결되어 중앙처리장치 내외부에 필요한 각종 제어신호를 출력하는 제어 유니트로 구성된 중앙처리장치에 있어서,

호스트 컴퓨터와 데이터 통신을 위한 데이터 통신수단;

중앙처리장치의 동작모드가 일반 동작 상태를 나타내는 일반 동작 모드인지 디버깅 상태를 나타내는 디버깅 모드인지를 나타내는 1 이상의 플래그들을 구비한 상태 레지스터;

메모리상의 디버깅 스택 영역을 지정하는 디버깅 스택 포인터 레지스터; 및

상기의 중앙처리장치를 초기화시키는 리셋 신호에 의해 리셋 상태로 전환되어 상기의 중앙처리장치를 초기화하고 리셋 상태에서 벗어날 때 디버깅 모드 진입 신호를 수신하여 상기의 중앙처리장치의 동작 모드를 디버깅 모드로 설정하는 제어 유닛을 구비하여 상기의 리셋 신호에 의해 리셋 상태로 전환되어 상기의 중앙처리장치를 초기화하고 리셋 상태에서 벗어날 때 디버깅 모드 진입 신호가 활성화되면 상기의 중앙처리장치를 디버깅 모드로 전환시키고 상기의 1이상의 플래그들이 디버깅 모드를 나타내는 값을 가지고 상기의 디버깅 스택 포인터 레지스터를 디버깅 프로그램의 데이터를 저장하는 메모리를 지정하는 스택 포인터로 사용하며 상기의 프로그램 카운터에는 디버깅 프로그램을 수행하기 위한 시작 주소를 적재하고 상기의 데이터 통신수단을 이용하여 호스트 컴퓨터로부터의 명령에 따라 디버깅을 수행하는 것을 특징으로 하는 프로그램 테스트 및 디버깅이 용이한 중앙처리장치.

【청구항 19】

제18항에 있어서, 상기의 프로그램 테스트 및 디버깅이 용이한 중앙처리장치는

설정된 리셋 데이터가 저장된 리셋 데이터 저장수단; 및

상기의 데이터 통신수단을 통해 입력받은 데이터와 상기의 리셋 데이터 저장수단에 저장된 설정된 리셋 데이터를 비교하여 상기의 데이터 통신수단을 통해 입력받은 데이

터와 설정된 리셋 데이터와 같으면 상기의 중앙처리장치를 초기화하는 리셋 데이터 비교 수단을 구비한 것을 특징으로 하는 프로그램 테스트 및 디버깅이 용이한 중앙처리장치.

【청구항 20】

제18항에 있어서, 디버깅 모드로 전환되었을 때 수행되는 디버깅 프로그램에서 사용하는 데이터 값을 저장하는 데이터 저장 메모리를 더 구비하여 일반 프로그램에서 사용하는 데이터 저장 메모리를 분리한 것을 특징으로 하는 프로그램 테스트 및 디버깅이 용이한 중앙처리장치.

【청구항 21】

제18항에 있어서, 디버깅 모드로 전환되었을 때 수행되는 디버깅 프로그램을 저장하는 메모리를 더 구비하여 일반 프로그램이 저장된 메모리와 분리한 것을 특징으로 하는 프로그램 테스트 및 디버깅이 용이한 중앙처리장치.

【청구항 22】

제18항에 있어서, 상기의 데이터 통신 수단을 통해 상기의 호스트 컴퓨터로부터 테스트 및 디버깅을 하기 위한 디버깅 프로그램을 다운로드 받는 것을 특징으로 하는 프로그램 테스트 및 디버깅이 용이한 중앙처리장치.

【청구항 23】

제18항에 있어서, 상기의 데이터 통신 수단을 통해 상기의 호스트 컴퓨터로부터 테스트 및 디버깅 할 응용 프로그램을 다운로드 받아 메모리에 저장하고 테스트 및 디버깅 할 응용 프로그램을 테스트 및 디버깅하는 것을 특징으로 하는 프로그램 테스트 및 디버깅이 용이한 중앙처리장치.

【청구항 24】

제18항에 있어서, 디버깅 모드로 전환되었을 때 상기의 프로그램 카운터에 저장된 값과 상기의 상태 레지스터에 저장된 데이터를 상기 디버깅 스택 포인터 레지스터가 지정하는 메모리에 저장하는 것을 특징으로 하는 프로그램 테스트 및 디버깅이 용이한 중앙처리장치.

【청구항 25】

제18항에 있어서, 임시 저장 레지스터를 더 구비하여 디버깅 모드로 전환되었을 때 상기의 프로그램 카운터에 저장된 값과 상기의 상태 레지스터에 저장된 데이터를 상기 임시 저장 레지스터에 저장하는 것을 특징으로 하는 프로그램 테스트 및 디버깅이 용이한 중앙처리장치.

【청구항 26】

데이터 및 어드레스 연산에 필요한 데이터를 일시적으로 저장하는 범용 레지스터 파일, 프로그램이 저장된 메모리의 주소를 기억하는 프로그램 카운터와 중앙처리장치의 상태를 나타내는 상태 레지스터 및 브레이크 레지스터로 구성된 특수 레지스터 파일, 상기의 범용 레지스터 파일 및 특수 레지스터 파일을 연결하는 내부 버스, 상기의 내부 버스에 연결되어 중앙처리장치 내외부에 필요한 각종 제어신호를 출력하는 제어 유니트로 구성된 중앙처리장치에 있어서,

호스트 컴퓨터와 데이터 통신을 위한 데이터 통신수단;

중앙처리장치의 동작모드가 일반 동작 상태를 나타내는 일반 동작 모드인지 디버깅 상태를 나타내는 디버깅 모드인지를 나타내는 1 이상의 플래그들을 구비한 상태 레지스

터;

메모리상의 디버깅 스택 영역을 지정하는 디버깅 스택 포인터 레지스터;

기준 데이터를 저장하는 기준 데이터 저장수단; 및

상기의 데이터 통신수단을 통해 입력받은 데이터와 상기의 기준 데이터 저장수단에 저장된 기준데이터를 비교하여 상기의 데이터 통신수단을 통해 입력받은 데이터와 기준 데이터가 동일하면 상기의 중앙처리장치를 디버깅 모드로 진입시키는 기준 데이터 비교 수단을 구비하여 상기의 데이터 통신수단을 통해 입력받은 데이터와 설정된 중지 데이터가 일치하면 상기의 중앙처리장치를 디버깅 모드로 전환시키고 상기의 1이상의 플래그들이 디버깅 모드를 나타내는 값을 가지고 상기의 디버깅 스택 포인터 레지스터를 디버깅 프로그램의 데이터를 저장하는 메모리를 지정하는 스택 포인터로 사용하며 상기의 프로그램 카운터에는 디버깅 프로그램을 수행하기 위한 시작 주소를 적재하고 상기의 데이터 통신수단을 이용하여 호스트 컴퓨터로부터의 명령에 따라 디버깅을 수행하는 것을 특징으로 하는 프로그램 테스트 및 디버깅이 용이한 중앙처리장치.

【청구항 27】

제26항에 있어서, 상기의 프로그램 테스트 및 디버깅이 용이한 중앙처리장치는

설정된 리셋 데이터가 저장된 리셋 데이터 저장수단; 및

상기의 데이터 통신수단을 통해 입력받은 데이터와 상기의 리셋 데이터 저장수단에 저장된 설정된 리셋 데이터를 비교하여 상기의 데이터 통신수단을 통해 입력받은 데이터와 설정된 리셋 데이터와 같으면 상기의 중앙처리장치를 초기화하는 리셋 데이터 비교 수단을 구비한 것을 특징으로 하는 프로그램 테스트 및 디버깅이 용이한 중앙처리장치.

【청구항 28】

제26항에 있어서, 상기의 제어 유닛은 디버깅 메모리 선택신호를 수신하여 상기의 브레이크 레지스터에 저장된 값과 상기의 중지데이터가 동일할 때 상기의 디버깅 메모리 선택신호에 따라 상기의 프로그램 카운터에 서로 다른 디버깅 프로그램을 수행하기 위한 주소를 적재하는 것을 특징으로 하는 프로그램 테스트 및 디버깅이 용이한 중앙처리장치.

【청구항 29】

제26항에 있어서, 디버깅 모드 전환되었을 때 수행되는 디버깅 프로그램에서 사용하는 데이터 값을 저장하는 데이터 저장 메모리를 더 구비하여 일반 프로그램에서 사용하는 데이터 저장 메모리를 분리한 것을 특징으로 하는 프로그램 테스트 및 디버깅이 용이한 중앙처리장치.

【청구항 30】

제26항에 있어서, 디버깅 모드로 전환되었을 때 수행되는 디버깅 프로그램을 저장하는 메모리를 더 구비하여 일반 프로그램이 저장된 메모리와 분리한 것을 특징으로 하는 프로그램 테스트 및 디버깅이 용이한 중앙처리장치.

【청구항 31】

제26항에 있어서, 상기의 데이터 통신 수단을 통해 상기의 호스트 컴퓨터로부터 테스트 및 디버깅을 하기 위한 디버깅 프로그램을 다운로드 받는 것을 특징으로 하는 프로그램 테스트 및 디버깅이 용이한 중앙처리장치.

【청구항 32】

제26항에 있어서, 상기의 데이터 통신 수단을 통해 상기의 호스트 컴퓨터로부터 테스트 및 디버깅 할 응용 프로그램을 다운로드 받아 메모리에 저장하고, 테스트 및 디버깅 할 응용 프로그램을 테스트 및 디버깅하는 것을 특징으로 하는 프로그램 테스트 및 디버깅이 용이한 중앙처리장치.

【청구항 33】

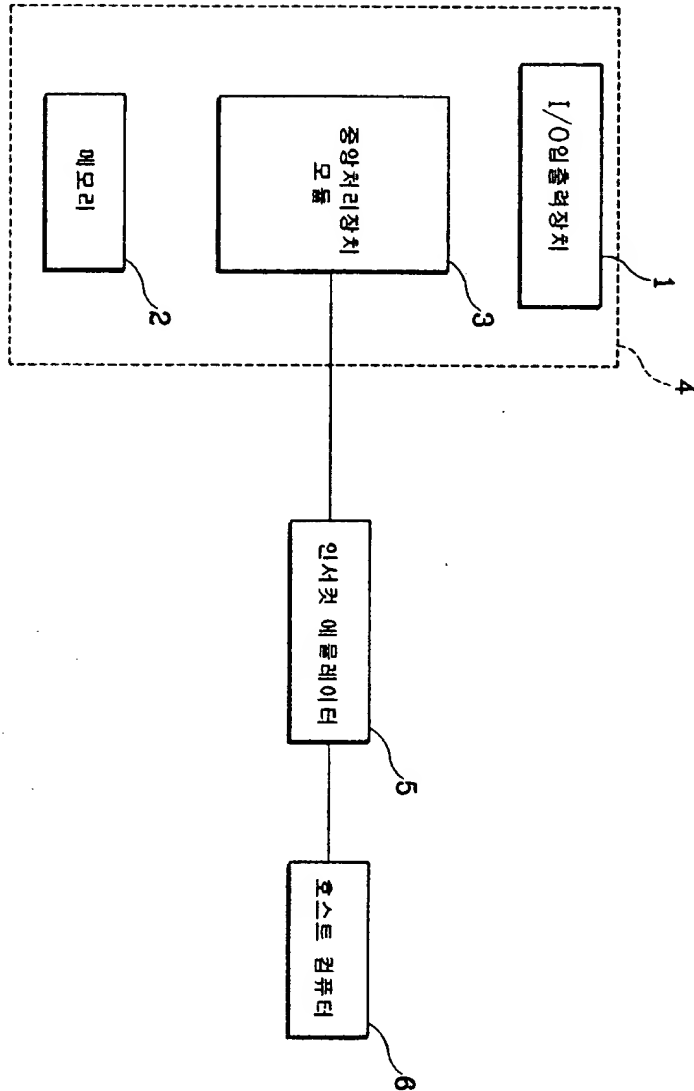
제26항에 있어서, 디버깅 모드로 전환되었을 때 상기의 프로그램 카운터에 저장된 값과 상기의 상태 레지스터에 저장된 데이터를 상기 디버깅 스택 포인터 레지스터가 지정하는 메모리에 저장하는 것을 특징으로 하는 프로그램 테스트 및 디버깅이 용이한 중앙처리장치.

【청구항 34】

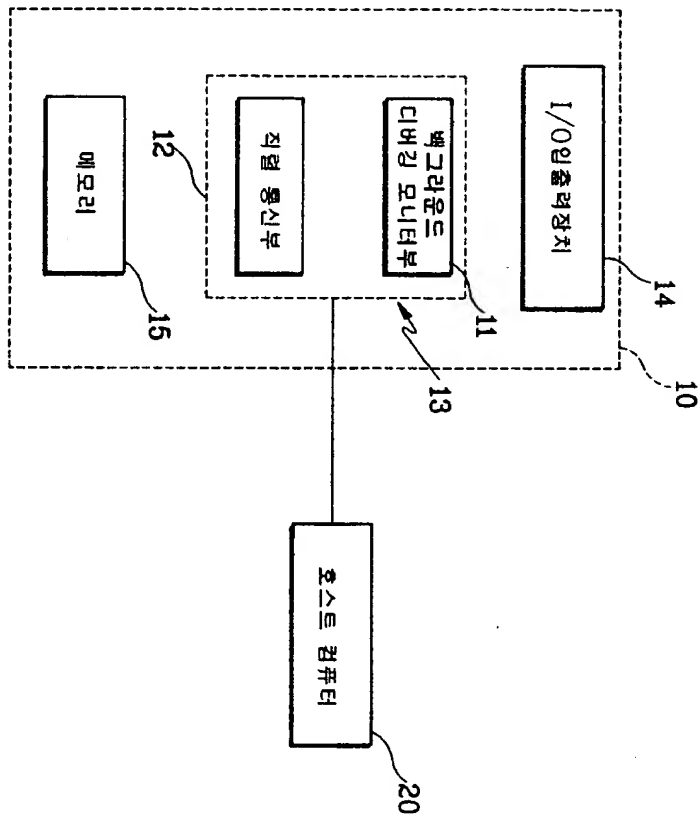
제26항에 있어서, 임시 저장 레지스터를 더 구비하여 디버깅 모드로 전환되었을 때 상기의 프로그램 카운터에 저장된 값과 상기의 상태 레지스터에 저장된 데이터를 상기 임시 저장 레지스터에 저장하는 것을 특징으로 하는 프로그램 테스트 및 디버깅이 용이한 중앙처리장치.

【도면】

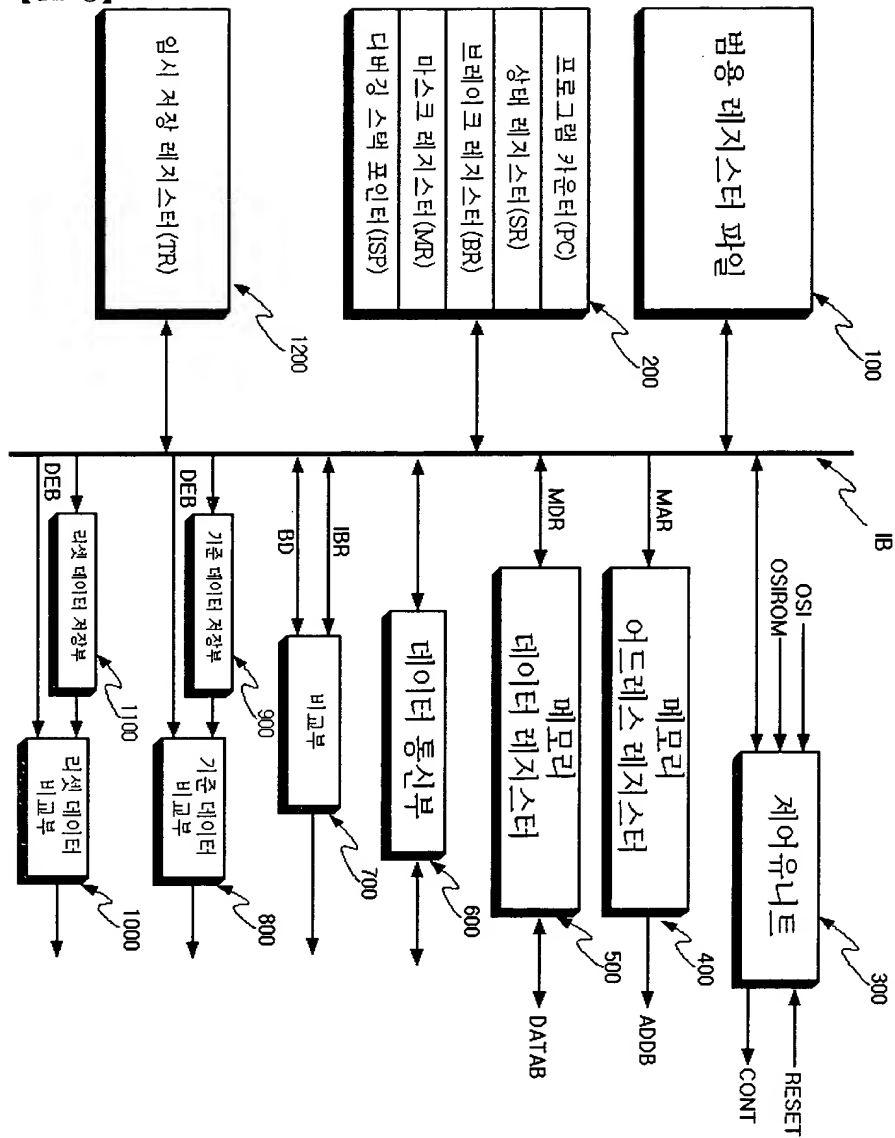
【도 1】



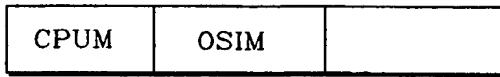
【도 2】



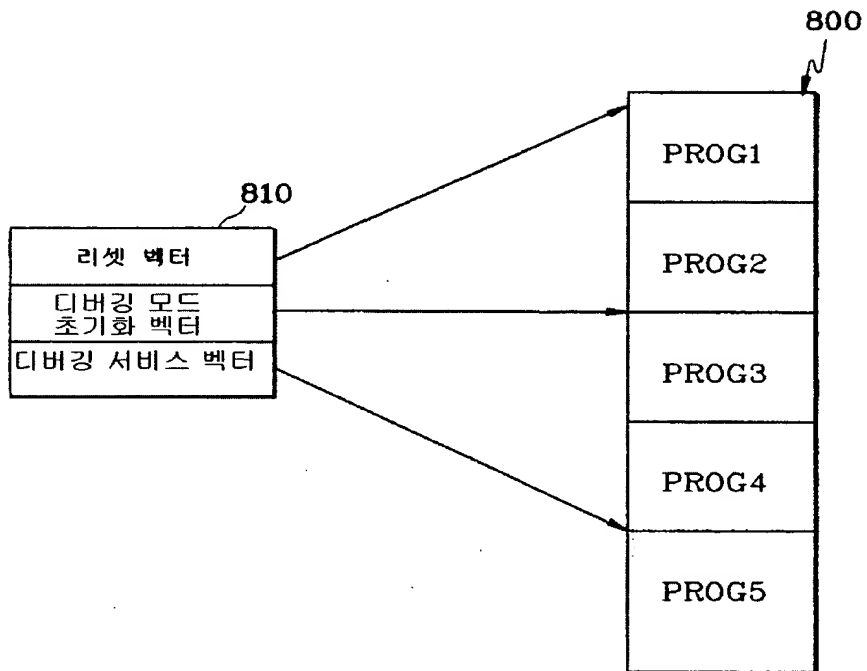
【나 3】



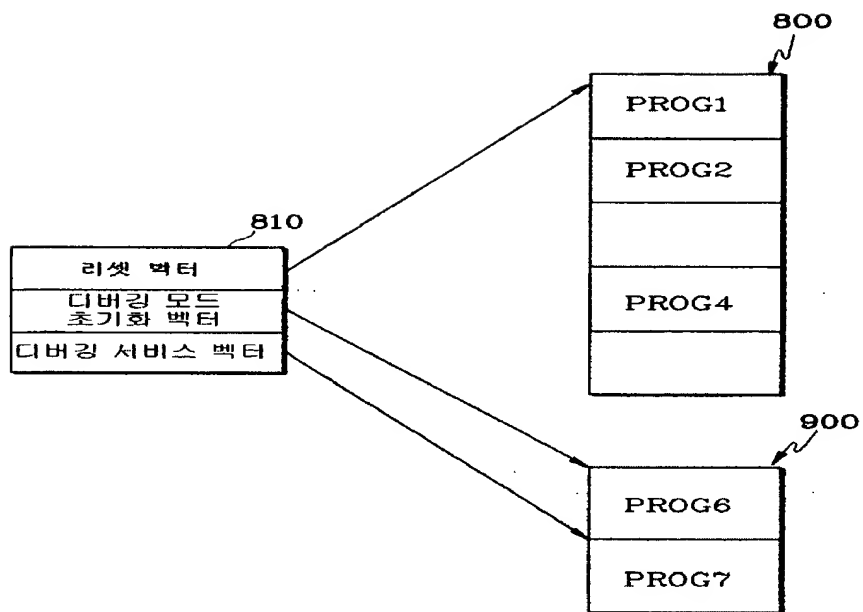
【도 4】



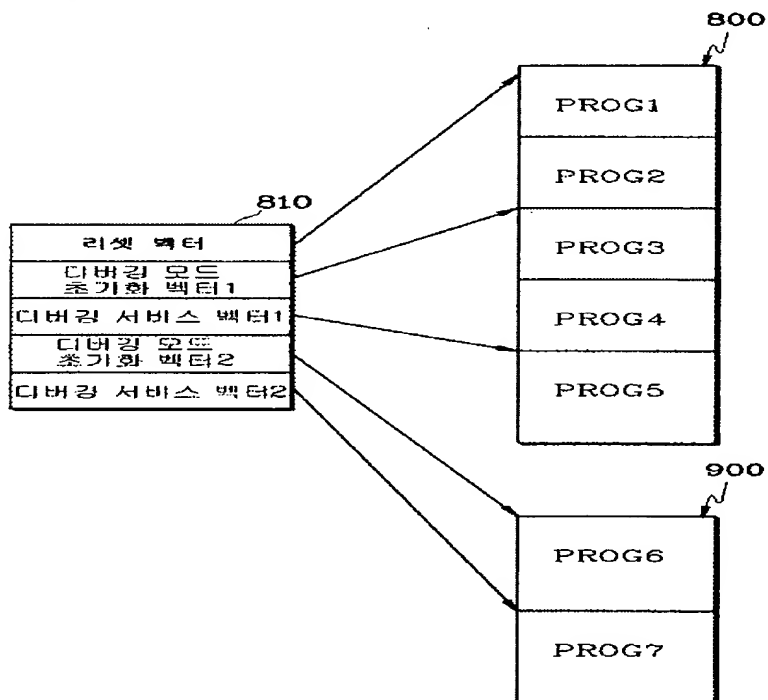
【도 5a】



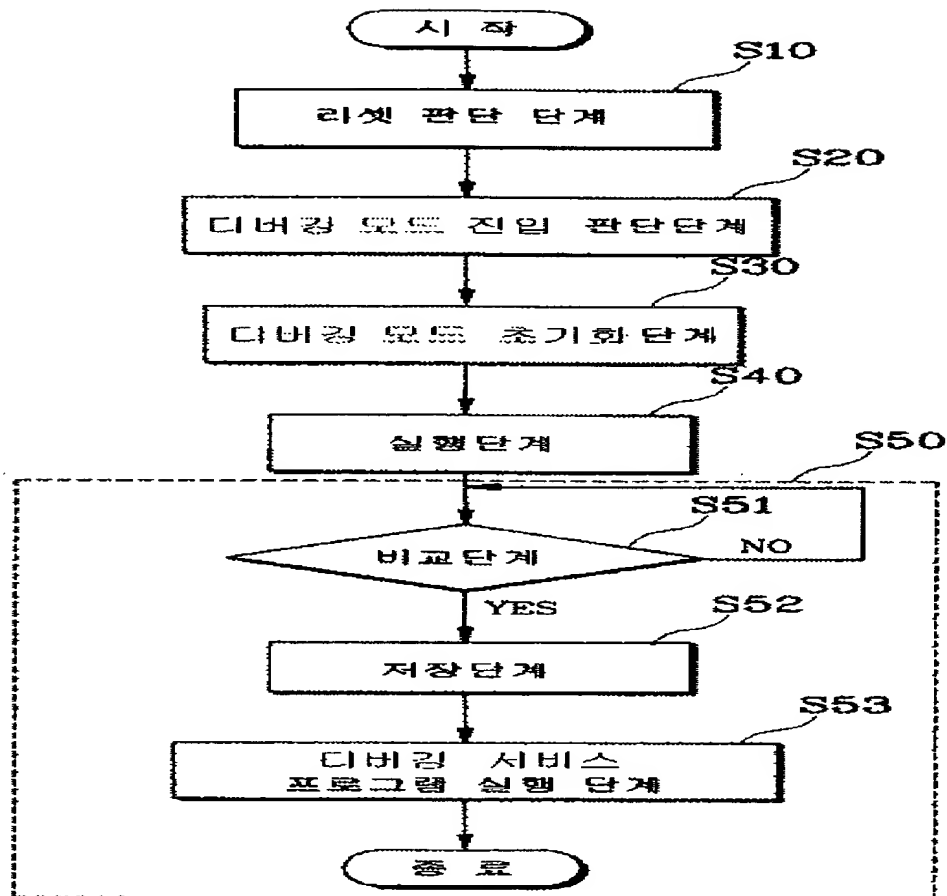
【도 5b】



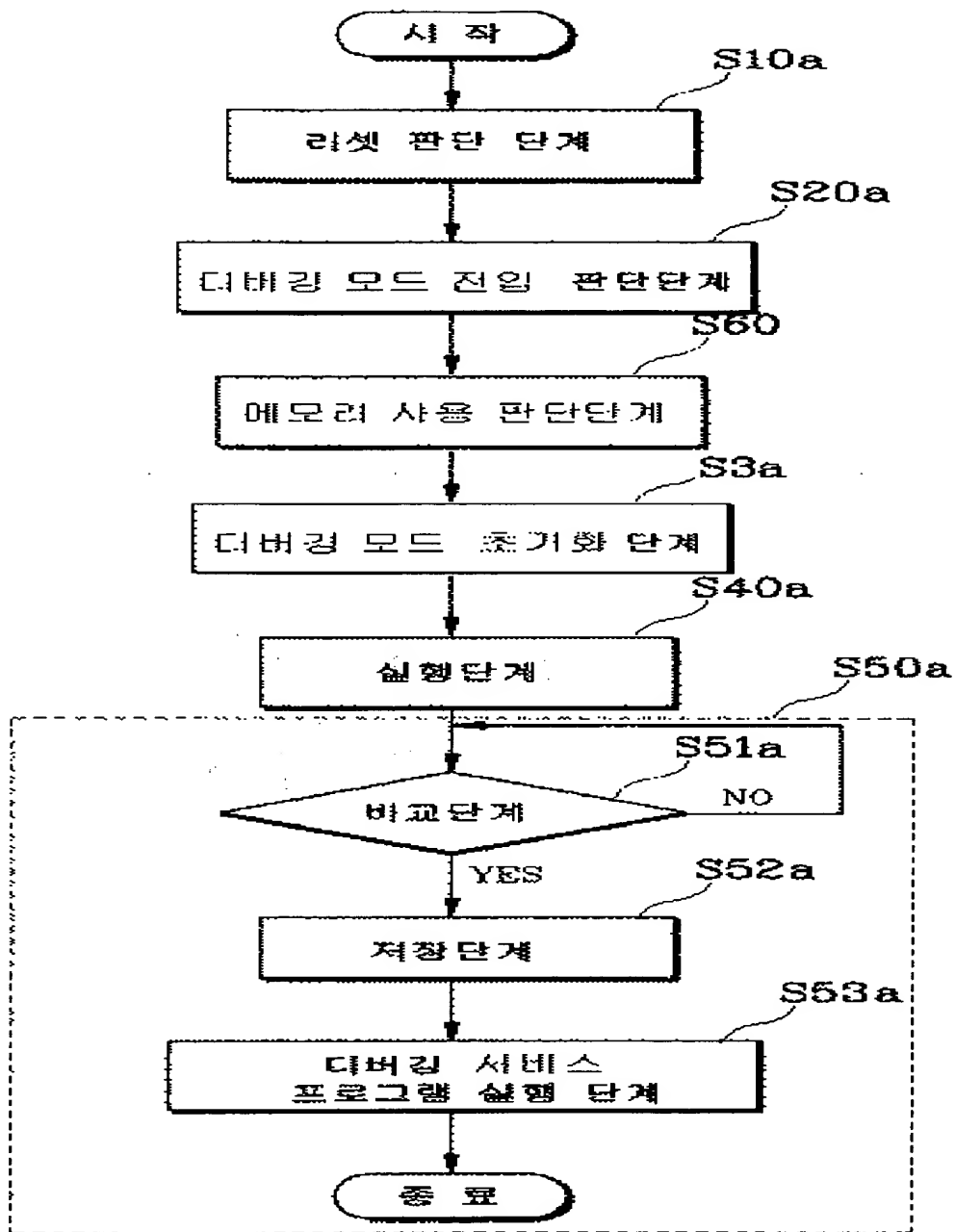
【도 5c】



【도 6】



【도 7】



【서류명】	명세서 등 보정서
【수신처】	특허청장
【제출일자】	2001.01.30
【제출인】	
【명칭】	주식회사 아시아디자인
【출원인코드】	1-1998-603147-8
【사건과의 관계】	출원인
【대리인】	
【성명】	황선웅
【대리인코드】	9-1999-000497-1
【포괄위임등록번호】	1999-068997-5
【사건의 표시】	
【출원번호】	10-2000-0038161
【출원일자】	2000.07.05
【심사청구일자】	2000.07.05
【발명의 명칭】	프로그램 테스트 및 디버깅이 용이한 중앙처리장치
【제출원인】	
【접수번호】	1-1-00-0139467-15
【접수일자】	2000.07.05
【보정할 서류】	명세서등
【보정할 사항】	
【보정대상 항목】	별지와 같음
【보정방법】	별지와 같음
【보정내용】	별지와 같음
【취지】	특허법시행규칙 제13조의 규정에 의하여 위와 같이 제출합니다. 대리인 황선웅 (인)
【수수료】	
【보정료】	0 원
【추가심사청구료】	0 원
【기타 수수료】	0 원
【합계】	0 원
【첨부서류】	1. 기타첨부서류_1통[보정사항]

【보정대상항목】 요약

【보정방법】 정정

【보정내용】

본 발명의 프로그램 테스트 및 디버깅이 용이한 중앙처리장치는 호스트 컴퓨터와 데이터 통신을 위한 데이터 통신부; 중앙처리장치의 동작모드가 일반 동작 상태를 나타내는 일반 동작 모드인지 디버깅 상태를 나타내는 디버깅 모드인지를 나타내는 플래그를 구비한 상태 레지스터; 메모리상의 디버깅 스택 영역을 지정하는 디버깅 스택 포인터 레지스터; 브레이크 레지스터에 저장된 값과 중지데이터를 비교하여 동일하면 중앙처리장치를 디버깅 모드로 진입시키는 비교부를 구비하여 상기의 브레이크 레지스터에 저장된 값과 중지 데이터가 일치하면 상기의 중앙처리장치를 디버깅 모드로 전환시키고 상기의 플래그들이 디버깅 모드를 나타내는 값을 가지며 상기의 디버깅 스택 포인터 레지스터를 디버깅 프로그램의 데이터를 저장하는 메모리를 지정하는 스택 포인터로 사용하며 데이터 통신수단을 통해 호스트 컴퓨터로부터의 명령에 따라 디버깅을 수행하는 것을 특징으로 한다.

【보정대상항목】 식별번호 7

【보정방법】 정정

【보정내용】

도 5c는 디버깅 프로그램이 일반 동작을 위한 응용 프로그램이 저장된 메모리 및 다른 독립된 메모리에 모두 저장된 경우 벡터 처리의 동작도이다.

【보정대상항목】 식별번호 8

【보정방법】 삭제

【보정대상항목】 식별번호 9

【보정방법】 삭제

【보정대상항목】 식별번호 10

【보정방법】 정정

【보정내용】

본 발명은 프로그램 테스트 및 디버깅이 용이한 중앙처리장치에 관한 것으로, 특히 디버깅 상태에서 호스트 컴퓨터로부터 다운로드 된 테스트 및 디버깅을 하기 위한 응용 프로그램을 용이하게 테스트 및 디버깅을 수행할 수 있는 프로그램 테스트 및 디버깅이 용이한 중앙처리장치에 관한 것이다.

【보정대상항목】 식별번호 12

【보정방법】 정정

【보정내용】

메모리에 저장된 다수의 명령어들의 집합으로 이루어진 프로그램이 올바르게 작성되었는지를 테스트하여야 하고, 응용 프로그램이 올바르게 작성되어 있지 않은 경우 그 응용 프로그램을 디버깅(Debugging)하여야 한다.

【보정대상항목】 식별번호 15

【보정방법】 정정

【보정내용】

인서킷 에뮬레이터(5)는 호스트 컴퓨터(6)로부터 응용 프로그램을 입력받아 응용 프로그램을 디버깅할 수 있는 디버깅용 프로그램을 저장하고 있으며, 호스트 컴퓨터(6)로부터 입력되는 명령에 따라 인서킷 에뮬레이터(5) 내부의 중앙처리장치를 동작시킨다. 그리고 중앙처리장치 모듈(3)을 통해 대상 시스템(4)의 제어 버스, 어드레스 버스 및 데이터 버스를 직접적으로 제어하여 대상 시스템(4)의 입출력장치(2) 및 메모리(3)에 필요한 입출력을 수행할 수 있고, 인서킷 에뮬레이터(5) 내부의 중앙처리장치의 내부 레지스터를 읽거나 내부 레지스터의 데이터를 변경할 수 있다.

【보정대상항목】 식별번호 16

【보정방법】 정정

【보정내용】

즉, 인서킷 에뮬레이터(5)는 호스트 컴퓨터(6)로부터 디버깅을 하기 위한 응용 프로그램을 입력받아 프로그램을 수행시켜 인서킷 에뮬레이터(5) 내의 중앙처리장치를 동작시킨다. 그리고 대상 시스템(4)의 제어 버스, 어드레스 버스 및 데이터 버스를 직접적으로 제어하여 입출력장치(2) 및 메모리(3)에 필요한 입출력을 수행하고, 응용 프로그램의 테스트 및 디버깅을 위하여 강제적으로 응용 프로그램의 수행을 중지시킨 후 호스트 컴퓨터(6)로부터의 명령에 따라 대상 시스템(4)의 입출력장치(2) 및 메모리(3)의 데이터를 읽거나 쓰고 또는 인서킷 에뮬레이터(5) 내의 레지스터에 저장된 데이터 값을 읽거나

써서 응용 프로그램이 올바르게 작성되었는지를 판단하고, 오류가 있는 경우 응용 프로그램을 디버깅한다.

【보정대상항목】 식별번호 17

【보정방법】 정정

【보정내용】

상기 종래의 인서킷 에뮬레이터를 사용하여 응용 프로그램의 테스트 및 디버깅을 하는 경우 반드시 대상 시스템의 중앙처리장치를 인서킷 에뮬레이터에 연결된 중앙처리장치 모듈로 대체될 수 있는 경우에만 유용하게 사용할 수 있고, 인서킷 에뮬레이터내에는 대상 시스템의 중앙처리장치의 기능을 하는 하드웨어가 있어야 하므로 인서킷 에뮬레이터는 고가의 장비라는 문제점을 가지고 있다. 그리고 대상 시스템을 하나의 반도체 칩으로 집적화 시킨 경우 중앙처리장치를 중앙처리장치 모듈(3)로 대체할 수 없으므로 인서킷 에뮬레이터를 사용할 수 없는 문제점을 가지고 있다.

【보정대상항목】 식별번호 19

【보정방법】 정정

【보정내용】

도 2의 블록디아그램은 도 1의 대상 시스템을 하나의 반도체 칩으로 집적시킨 경우 인서킷 에뮬레이터를 사용할 수 없는 문제점을 해결하기 위해 중앙처리장치는 백그라운드 디버깅 모니터부(Background Debugging Monitor : BDM)(11)와 직렬 통신부(12)를 별도로 가지고 있다.

【보정대상항목】 식별번호 20

【보정방법】 정정

【보정내용】

도 2의 종래의 중앙처리장치 내에 백그라운드 디버깅 모니터부(11)가 내장되어 프로그램의 테스트 및 디버깅을 하기 위한 블럭다이어그램은 별도의 백그라운드 디버깅 모니터부(11) 및 직렬 통신부(12)를 가진 중앙처리장치(13), 입출력장치(14) 및 메모리(15)가 하나의 반도체 칩으로 집적화된 대상 시스템(10)과 직렬 통신부(12)를 통해 응용 프로그램의 테스트 및 디버깅을 하기 위한 호스트 컴퓨터(20)로 구성된다.

【보정대상항목】 식별번호 21

【보정방법】 정정

【보정내용】

백그라운드 디버깅 모니터부(11)는 호스트 컴퓨터(20)로부터 직렬 통신부(12)를 통해 입력된 명령어를 해석하여 필요한 동작을 수행하기 위해 자체적으로 레지스터, 제어 유닛 등으로 구성된 컨트롤러로 중앙처리장치(13) 내부의 모든 구성요소들을 직접적으로 제어할 수 있어야 한다. 즉, 백그라운드 디버깅 모니터부(11)는 내부 버스에 연결되어 레지스터를 읽거나 값을 변경할 수 있다. 그리고 제어부, 메모리 어드레스 레지스터 및 메모리 데이터 레지스터를 제어하여 제어신호버스, 어드레스 버스 및 데이터 버스를 제어한다.

【보정대상항목】 식별번호 24

【보정방법】 정정

【보정내용】

본 발명의 목적은 디버깅 입력신호를 활성화시켜 사용자 모드를 갖는 중앙처리장치의 상태에 디버깅 모드를 새로이 추가하고, 디버깅 모드에서 호스트 컴퓨터로부터 다운로드 된 테스트 및 디버깅을 하기 위한 응용 프로그램을 용이하게 테스트 및 디버깅함으로써 중앙처리장치의 하드웨어가 단순하게 구성되어 중앙처리장치의 가격을 하락시킬 수 있고, 보다 효율적인 테스트 및 디버깅을 할 수 있는 프로그램 테스트 및 디버깅이 용이한 중앙처리장치를 제공하는 데 있다.

【보정대상항목】 식별번호 25

【보정방법】 정정

【보정내용】

상기의 목적을 달성하기 위하여 본 발명의 프로그램 테스트 및 디버깅이 용이한 중앙처리장치는 호스트 컴퓨터와 데이터 통신을 위한 데이터 통신부; 중앙처리장치의 동작모드가 일반 동작 상태를 나타내는 일반 동작 모드인지 디버깅 상태를 나타내는 디버깅 모드인지를 나타내는 플래그를 구비한 상태 레지스터; 디버깅 프로그램의 데이터를 저장하는 스택 메모리를 지정하는 스택 포인터로 사용되는 디버깅 스택 포인터 레지스터; 및 브레이크 레지스터에 저장된 값과 중지데이터를 비교하여 동일하면 중앙처리장치를 디버깅 모드로 진입시키는 비교부를 구비하여 브레이크 레지스터에 저장된 값과 중지 데이터가 일치하면 중앙처리장치를 디버깅 모드로 전환시키고 상태 레지스터의 플래그가

디버깅 모드를 나타내는 값을 가지고 프로그램 카운터에 디버깅 프로그램을 수행하기 위한 시작 주소를 적재하여 디버깅 프로그램을 수행하여 데이터 통신부를 통해 호스트 컴퓨터로부터의 명령에 따라 디버깅을 수행하는 것을 특징으로 한다.

【보정대상항목】 식별번호 27

【보정방법】 정정

【보정내용】

이하, 첨부된 도면을 참조하여 본 발명의 프로그램 테스트 및 디버깅이 용이한 중앙처리장치를 상세히 설명하고자 한다.

【보정대상항목】 식별번호 29

【보정방법】 정정

【보정내용】

도 3의 본 발명의 프로그램 테스트 및 디버깅이 용이한 중앙처리장치는 데이터 및 어드레스 연산에 필요한 데이터를 일시적으로 저장하는 범용 레지스터 파일(100), 프로그램이 저장된 메모리의 주소를 기억하는 프로그램 카운터(PC)와 중앙처리장치의 일반 동작 상태를 나타내는 일반 동작 모드 및 프로그램 테스트 및 디버깅을 하기 위한 상태를 나타내는 디버깅 모드를 나타내는 디버깅 모드 플래그를 가진 상태 레지스터(SR)와 브레이크 레지스터(BR), 마스크 레지스터(MR) 및 디버깅 초기화 프로그램과 디버깅 서비스 프로그램의 데이터를 저장하는 메모리를 지정하는 디버깅 스택 포인터(ISP)로 구성된 특수 레지스터 파일(200), 범용 레지스터 파일(100) 및 특수 레지스터 파일(200)을 연결하는 내부 버스(IB), 내부 버스(IB)에 연결되어 중앙처리장치 내외부에 필요한 각종

제어신호(CONT)를 출력하고 리셋 신호(RESET)에 의해 중앙처리장치를 초기화하고 디버깅 모드 진입 신호(OSI)를 검사하여 디버깅 모드 진입 신호(OSI)가 활성화되면 프로그램 카운터(PC)에 디버깅 초기화 프로그램을 수행하기 위한 시작주소를 적재하여 중앙처리장치를 디버깅 초기화 모드로 전환시키고 상태 레지스터(SR)의 플래그는 디버깅 모드를 나타내는 값을 가지도록 하는 제어유닛(300), 메모리 어드레스 레지스터(400), 메모리 데이터 레지스터(500), 호스트 컴퓨터와 데이터 통신을 위한 데이터 통신부(600) 및 브레이크 레지스터(BR)에 저장된 값(IBR)과 중지데이터(BD)를 비교하여 동일하면 중앙처리장치를 디버깅 모드로 진입시키는 비교부(700)로 구성된다.

또한 본 발명의 프로그램 테스트 및 디버깅이 용이한 중앙처리장치는 호스트 컴퓨터와 데이터 통신을 위한 데이터 통신부(600), 중앙처리장치의 동작모드가 일반 동작 상태를 나타내는 일반 동작 모드인지 디버깅 상태를 나타내는 디버깅 모드인지를 나타내는 플래그를 구비한 상태 레지스터(SR), 디버깅 프로그램의 데이터를 저장하는 스택 메모리를 지정하는 스택 포인터로 사용되는 디버깅 스택 포인터(ISP) 및 브레이크 레지스터에 저장된 값과 중지데이터를 비교하여 동일하면 중앙처리장치를 디버깅 모드로 진입시키는 비교부(700)로 구성되어 초기 디버깅 모드 진입에 의해 호스트 컴퓨터로부터 입력된 브레이크 레지스터(BR)에 저장된 값과 중지 데이터(BD)가 일치하면 프로그램 카운터(PC)에 디버깅 초기화 프로그램과 디버깅 서비스 프로그램으로 이루어진 디버깅 프로그램을 수행하기 위한 시작 주소를 적재하여 디버깅 프로그램을 수행하여 중앙처리장치를 디버깅 모드로 전환시키고 상태 레지스터의 플래그가 디버깅 모드를 나타내는 값을 가지고 데이터 통신부(600)를 통해 호스트 컴퓨터로부터의 명령에 따라 디버깅을 수행할 수 있다.

비교부(700)에 입력되는 중지데이터(BD)는 프로그램 카운터(PC)에 저장된 프로그램 번

지, 데이터가 저장된 메모리 어드레스, 입출력되는 데이터, 입출력되는 어드레스일 수 있고, 마스크 레지스터(MR)를 사용하여 프로그램 카운터에 저장된 프로그램 번지와 마스크 레지스터(MR)에 저장된 값을 연산한 결과, 데이터가 저장된 메모리 어드레스와 마스크 레지스터(MR)에 저장된 값을 연산한 결과, 입출력되는 어드레스와 마스크 레지스터(MR)에 저장된 값을 연산한 결과, 입출력되는 데이터와 마스크 레지스터(MR)에 저장된 값을 연산한 결과일 수 있다.

도 3, 도 5a 및 도 5b에 도시된 바와 같이 제어유니트(300)는 디버깅 메모리 선택신호(OSIROM)를 수신하여 브레이크 레지스터(BR)에 저장된 값(IBR)과 중지데이터(BD)가 동일할 때 디버깅 메모리 선택신호(OSIROM)에 따라 프로그램 카운터(PC)에 디버깅 서비스 프로그램(PROG5, PROG7)을 수행하기 위한 서로 다른 주소를 적재할 수 있다. 또한 도 5b에 도시된 바와 같이 디버깅 초기화 프로그램(PROG6)과 디버깅 서비스 프로그램(PROG7)으로 구성된 디버깅 프로그램을 저장하는 메모리(90)와 일반 프로그램이 저장된 메모리(80)를 분리할 수 있다.

디버깅 프로그램에서 사용하는 데이터 값을 저장하는 데이터 저장 메모리와 일반 프로그램에서 사용하는 데이터 값을 저장하는 데이터 저장 메모리를 분리할 수 있다.

디버깅 모드로 전환되었을 때 본 발명의 프로그램 테스트 및 디버깅이 용이한 중앙처리장치는 빠른 디버깅을 위하여 임시 저장 레지스터(1200)를 더 구비하여 디버깅 모드로 전환한 뒤 디버깅 프로그램을 수행하기 전에 프로그램 카운터(PC)와 상태 레지스터(SR)를 임시 저장 레지스터(1200)에 저장하고 디버깅 프로그램을 수행 할 수도 있다.

상기 프로그램 테스트 및 디버깅이 용이한 중앙처리장치는 기준데이터가 저장된 기준 데이터 저장부(900) 및 데이터 통신부(600)를 통해 입력받은 데이터(DEB1)와 기준데이터

를 비교하여 데이터 통신부(600)를 통해 입력받은 데이터(DEB1)와 기준데이터가 동일하면 중앙처리장치를 디버깅 서비스 모드로 전환시키는 기준 데이터 비교부(800)를 구비하여 사용자가 호스트 컴퓨터를 이용하여 데이터 통신부를 통해 중앙처리장치를 강제로 디버깅 서비스 모드로 전환시켜 디버깅 서비스 프로그램을 수행시킬 수 있다.

본 발명의 프로그램 테스트 및 디버깅이 용이한 중앙처리장치는 미리 설정된 리셋 데이터를 저장하는 리셋 데이터 저장부(1100)와 리셋 데이터 저장부(1100)에 저장된 리셋데이터와 데이터 통신부(600)를 통해 입력된 디버깅 데이터(DEB2)를 비교하는 리셋 데이터 비교부(1000)를 더 구비하여, 사용자가 중앙처리장치를 강제로 초기화시키기 위하여 호스트 컴퓨터에서 데이터 통신부(600)로 데이터 전송하면 리셋 데이터 비교부(1000)에서 리셋 데이터 저장부(1100)에 저장된 리셋 데이터와 데이터 통신부(600)로 수신된 값을 비교하여 두 값이 동일하면 중앙처리장치를 초기화시킨다.

【보정대상항목】 식별번호 30

【보정방법】 삭제

【보정대상항목】 식별번호 31

【보정방법】 삭제

【보정대상항목】 식별번호 32

【보정방법】 삭제

【보정대상항목】 식별번호 33

【보정방법】 삭제

【보정대상항목】 식별번호 34

【보정방법】 삭제

【보정대상항목】 식별번호 35

【보정방법】 삭제

【보정대상항목】 식별번호 36

【보정방법】 삭제

【보정대상항목】 식별번호 37

【보정방법】 삭제

【보정대상항목】 식별번호 38

【보정방법】 삭제

【보정대상항목】 식별번호 39

【보정방법】 삭제

【보정대상항목】 식별번호 40

【보정방법】 삭제

【보정대상항목】 식별번호 41

【보정방법】 삭제

【보정대상항목】 식별번호 42

【보정방법】 삭제

【보정대상항목】 식별번호 43

【보정방법】 정정

【보정내용】

상기의 구성에 따른 본 발명인 프로그램의 테스트 및 디버깅이 용이한 중앙처리장치의 동작은 다음과 같다.

제어유니트(300)는 리셋 신호(RESET)에 의해 중앙처리장치를 초기화하고 제어유니트(300)로 입력되는 디버깅 모드 진입 신호(OSI)가 활성화되면 도 4에 도시된 바와 같이 상태 레지스터(SR)의 디버깅 모드 플래그(OSIM)를 활성화시켜, 중앙처리장치를 응용 프

로그래ムの 테스트 및 디버깅을 수행하기 위한 초기 환경으로 설정해 주고, 데이터 통신부(600)를 통해 호스트 컴퓨터로부터 중앙처리장치로 다운로드 된 테스트 및 디버깅을 하기 위한 응용 프로그램을 실행한다. 응용 프로그램 실행중에 비교부(700)에 의해 브레이크 레지스터(BR)에 저장된 값(IBR)과 중지데이터(BD)가 동일한지 판단하여, 두 값이 동일하면 제어유니트(300)는 중앙처리장치를 디버깅 모드로 진입시킨 후 데이터 통신부(600)를 통해 호스트 컴퓨터로부터의 명령에 따라 응용 프로그램의 테스트 및 디버깅을 수행한다.

응용 프로그램은 디버깅 모드 진입 신호(OSI)가 활성화되어 테스트 및 디버깅을 수행하기 위한 초기 환경으로 중앙처리장치를 설정한 후 호스트 컴퓨터로부터 중앙처리장치로 다운로드 받거나, 중앙처리장치를 테스트 및 디버깅을 수행하기 위한 초기 환경으로 설정한 후 비교부(700)에 의해 브레이크 레지스터(BR)에 저장된 값(IBR)과 중지데이터(BD)가 동일하면 중앙처리장치를 프로그램 테스트 및 디버깅 상태로 진입시킨 후 호스트 컴퓨터로부터 중앙처리장치로 다운로드 받을 수 있다.

비교부(700)에 의해 브레이크 레지스터(BR)에 저장된 값(IBR)과 중지데이터(BD)가 동일하면 제어유니트(300)는 응용 프로그램의 실행 중인 동작을 중지 또는 완료하고, 프로그램 카운터(PC)에 저장된 값과 상태 레지스터(SR)에 저장된 데이터를 임시 저장 레지스터(TR)나 디버깅 스택 포인터(ISP)가 지정하는 스택 메모리에 저장하고, 프로그램 카운터(PC)에는 응용 프로그램을 테스트 및 디버깅하기 위한 디버깅 서비스 프로그램을 수행하기 위한 시작 주소를 저장하고, 중앙처리장치를 응용 프로그램의 테스트 및 디버깅하기 위한 상태에 있도록 상태 레지스터(SR)의 데이터를 변경한다.

비교부(700)에 입력되는 중지데이터(BD)는 프로그램 카운터(PC)에 저장된 프로그램 번

지, 데이터가 저장된 메모리 어드레스, 입출력되는 데이터, 입출력되는 어드레스일 수 있고, 마스크 레지스터(MR)를 사용하여 프로그램 카운터에 저장된 프로그램 번지와 마스크 레지스터(MR)에 저장된 값을 연산한 결과, 데이터가 저장된 메모리 어드레스와 마스크 레지스터(MR)에 저장된 값을 연산한 결과, 입출력되는 어드레스와 마스크 레지스터(MR)에 저장된 값을 연산한 결과, 입출력되는 데이터와 마스크 레지스터(MR)에 저장된 값을 연산한 결과일 수 있다. 마스크 레지스터(MR)는 검사하고자 하는 데이터의 일부만을 비교하고자 할 때 유용하게 사용된다. 즉, 입출력되는 데이터 중 하위 6비트들만을 검사하고자 할 경우 마스크 레지스터(MR)에 16진수로 3F란 값을 설정하고 입출력되는 데이터와 마스크 레지스터(MR)를 논리곱 연산을 수행하여 중지데이터(BD)의 하위 6비트들과 브레이크 레지스터(BR)의 하위 6비트들이 동일하면 중앙처리장치를 디버깅 모드로 전환하도록 할 수 있다.

도 5a에 도시된 바와 같이 중앙처리장치의 초기화 및 일반 동작을 위한 응용 프로그램(PROG1, PROG2, PROG4), 디버깅 모드 진입 신호(OSI)의 활성화에 따라 중앙처리장치를 응용 프로그램의 테스트 및 디버깅을 수행하기 위한 초기 환경으로 설정해 주는 디버깅 모드 초기화 프로그램(PROG3)과 비교부(700)에 의해 중앙처리장치를 프로그램 테스트 및 디버깅 상태로 진입시켜 응용 프로그램을 테스트 및 디버깅하는 디버깅 서비스 프로그램(PROG5)은 동일한 메모리(80)에 저장되거나 도 5b에 도시된 바와 같이 디버깅 모드 초기화 프로그램(PROG6)과 디버깅 서비스 프로그램(PROG7)은 다른 독립된 메모리(90)에 저장될 수 있다.

도 5b에 도시된 바와 같이 일반 동작을 위한 응용 프로그램(PROG1, PROG2, PROG4)을 저장하는 메모리(80)와 디버깅 모드 초기화 프로그램(PROG6)과 디버깅 서비스 프로그램

(PROG7)을 저장하는 메모리(90)를 분리시키고, 메모리(90)를 중앙처리장치와 함께 집적화(integrated circuit) 함으로써 중앙처리장치의 동작속도를 빠르게 할 수 있다.

또는 도 5c에 도시된 바와 같이 디버깅 모드 초기화 프로그램(PROG6) 및 디버깅 서비스 프로그램(PROG7)이 저장되는 중앙처리장치와 함께 집적화 되는 메모리(90)와 별도로 사용자가 직접 디버깅 모드 초기화 프로그램(PROG3) 및 디버깅 서비스 프로그램(PROG5)을 작성하여 중앙처리장치를 초기화 및 일반 동작을 위한 응용 프로그램(PROG1, PROG2, PROG4)이 저장된 메모리(80)에 디버깅 모드 초기화 프로그램(PROG3) 및 디버깅 서비스 프로그램(PROG5)을 저장할 수도 있다.

【보정대상항목】 식별번호 44

【보정방법】 정정

【보정내용】

도 5a에 도시된 바와 같이 중앙처리장치를 초기화 및 일반 동작을 위한 응용 프로그램(PROG1, PROG2, PROG4), 디버깅 모드 초기화 프로그램(PROG3) 및 디버깅 서비스 프로그램(PROG5)이 모두 동일한 메모리(80)에 저장되어 있는 경우 본 발명의 동작을 상세히 설명하면 다음과 같다.

【보정대상항목】 식별번호 45

【보정방법】 정정

【보정내용】

중앙처리장치에 전원이 공급되면 리셋 신호(RESET)가 활성화된다. 리셋 신호(RESET)에 따라 제어유니트(300)는 중앙처리장치를 초기화하고, 디버깅 모드 진입 신호

(OSI)의 활성화 여부를 판단한다. 디버깅 모드 진입 신호(OSI)가 비활성화되어 있으면 도 4의 상태 레지스터(SR)의 디버깅 모드 플래그(OSIM)를 비활성화시킨다. 그리고 제어유닛(300)은 벡터 테이블(81)에서 최초로 수행할 프로그램이 저장된 메모리(80) 주소가 저장되어 있는 리셋 벡터를 읽어들이어 특수 레지스터 파일(200)의 프로그램 카운터(PC)에 저장한다. 제어유닛(300)은 프로그램 카운터(PC)에 저장된 리셋 벡터를 메모리 어드레스 레지스터(400)로 이동시키고 어드레스 버스(ADDB)로 출력한다. 그리고 메모리 읽기 신호를 제어버스(CONT)로 출력하여 응용 프로그램(PROG1)이 저장된 메모리(80) 내용을 읽어 들이게 된다. 메모리에서 출력된 데이터는 데이터 버스(DATAB) 및 메모리 데이터 레지스터(500)를 통하여 내부버스(IB)를 경유하여 제어유닛(300)로 들어오게 된다. 이렇게 제어유닛(300)로 입력된 명령어는 해석되어 제어유닛(300)로부터 필요한 제어신호들(CONT)이 출력되어 해당 명령어를 수행한다.

【보정대상항목】 식별번호 46

【보정방법】 정정

【보정내용】

리셋 벡터에 의해 디버깅하기 위한 응용 프로그램(PROG1)을 수행하기 위해 중앙처리장치가 초기화되고, 디버깅 모드 진입 신호(OSI)가 비활성화되어 있으면 중앙처리장치는 그 응용 프로그램(PROG1)을 수행한다.

【보정대상항목】 식별번호 47

【보정방법】 정정

【보정내용】

만일 리셋 신호(RESET)에 의해 중앙처리장치가 초기화되고, 제어유니트(300)로 입력되는 디버깅 모드 진입 신호(OSI)가 활성화되면, 제어유니트(300)는 상태 레지스터(SR)의 디버깅 모드 플래그(OSIM)를 활성화시킨다. 그리고 제어유니트(300)는 메모리(80)에 저장된 디버깅 모드 초기화 프로그램(PROG3)의 시작 주소인 디버깅 모드 초기화 벡터를 읽어들이어 프로그램 카운터(PC)에 저장한다. 프로그램 카운터(PC)에 저장된 디버깅 모드 초기화 벡터에 의해 디버깅 모드 초기화 프로그램(PROG3)을 수행한 후, 중앙처리장치는 응용 프로그램의 테스트 및 디버깅을 수행하기 위한 초기 환경으로 설정된다.

【보정대상항목】 식별번호 48

【보정방법】 삭제

【보정대상항목】 식별번호 49

【보정방법】 정정

【보정내용】

제어유니트(300)는 호스트 컴퓨터에서의 명령에 의해 상태 레지스터(SR)의 디버깅 모드 플래그(OSIM)를 비활성화시키고 수행할 응용 프로그램의 명령어가 저장되어 있는 주소를 프로그램 카운터(PC)에 저장하여 디버깅을 하기 위한 응용 프로그램을 실행한다.

응용 프로그램의 실행 중 비교부(700)에 의해 특수 레지스터 파일(200)의 브레이크 레지스터(BR)에 저장된 값(IBR)과 중지데이터(BD)가 동일하면, 중앙처리장치는 응용 프로그램의 실행 중인 동작을 중지 또는 완료하고, 프로그램 카운터(PC)에 저장된 응용 프로그램의 주소 값과 상태 레지스터(SR)에 저장된 데이터를 임시 저장 레지스터(TR)(1200)에 저장하고, 프로그램 카운터(PC)에는 응용 프로그램을 테스트 및 디버깅하기 위한 디버깅 서비스 프로그램(PROG5)의 시작 주소인 디버깅 서비스 벡터를 저장하고, 응용 프로그램을 테스트 및 디버깅하기 위한 상태에 있도록 중앙처리장치의 상태 레지스터(SR) 값을 변경한다. 즉 상태 레지스터(SR)의 디버깅 모드 플래그(OSIM)를 활성화시킨다.

【보정대상항목】 식별번호 51

【보정방법】 정정

【보정내용】

중앙처리장치는 프로그램 테스트 및 디버깅 상태로 진입한 후 메모리(80)에 저장된 디버깅 서비스 프로그램(PROG5)에 따라 호스트 컴퓨터로부터의 명령, 예를 들어 사용자가 작성한 응용 프로그램의 다운로드, 다운로드된 응용 프로그램의 실행, 브레이크 레지스터(BR)의 데이터 변경, 범용 레지스터 파일(100) 및 특수 레지스터 파일(200)의 레지스터들에 저장된 데이터의 판독(READ) 또는 기록(WRITE) 및 메모리에 저장된 데이터들의 판독 또는 기록 등의 명령을 수행한다. 이러한 명령들을 이용하여 사용자는 응용 프로그램이 올바르게 작성되었는지를 판단하고, 잘못 작성되었으면 이를 수정하여 바로 잡는다.

【보정대상항목】 식별번호 53

【보정방법】 정정

【보정내용】

호스트 컴퓨터로부터의 다운로드 명령에 의해 사용자가 작성한 프로그램을 데이터 통신부(600)를 통해 전송받아 중앙처리장치의 메모리에 저장한 후 중앙처리장치는 성공적으로 수신 받았다는 신호를 호스트 컴퓨터로 송신한다. 만약 응용 프로그램의 전송 중 에러가 발생했다면 중앙처리장치는 호스트 컴퓨터로 응용 프로그램의 재전송을 요구한다. 응용 프로그램을 중앙처리장치에 성공적으로 다운로드 한 후 디버깅을 위하여 호스트 컴퓨터는 중앙처리장치의 브레이크 레지스터(BR)의 데이터 값을 설정하기 위한 브레이크 레지스터(BR) 데이터 변경 명령을 내린다. 브레이크 레지스터(BR) 데이터 변경 명령 후 중앙처리장치는 호스트 컴퓨터로 성공적으로 수신 받았다는 신호를 송신한다. 호스트 컴퓨터는 중앙처리장치의 메모리에 다운로드된 응용 프로그램을 실행시키기 위하여 응용 프로그램의 실행 명령을 중앙처리장치로 송신하고, 중앙처리장치는 이에 응답하고 응용 프로그램을 실행한다. 응용 프로그램의 실행 중 비교부(700)에 의해 브레이크 레지스터(BR)의 데이터 값과 중지데이터인 프로그램 카운터(PC)의 데이터 값이 동일하면, 응용 프로그램의 실행 중인 동작을 중지 또는 완료하고 프로그램 카운터(PC)에 저장된 값과 상태 레지스터(SR)에 저장된 데이터 및 범용 레지스터의 내용들을 임시 저장 레지스터(TR) 또는 디버깅 스택 포인터(ISP)가 지정하는 스택에 저장한다. 프로그램 카운터(PC)에는 응용 프로그램을 테스트 및 디버깅하는 디버깅 서비스 프로그램(PROG5)의 시작 주소인 디버깅 서비스 벡터를 저장하고, 중앙처리장치를 응용 프로그램의 테스트 및 디버깅하기 위한 상태에 있도록 상태 레지스터(SR)의 데이터를 변경한다. 즉 상태 레지스

터(SR)의 데이터를 변경한다. 즉, 상태 레지스터(SR)의 디버깅 모드 플래그(OSIM)를 활성화시킨다. 디버깅 서비스 프로그램(PROG5)에 의해 중앙처리장치는 응용 프로그램의 테스트 및 디버깅을 수행하기 위한 상태로 진입된다. 중앙처리장치가 테스트 및 디버깅을 수행하기 위한 디버깅 모드로 진입한 후 호스트 컴퓨터로부터 레지스터 판독 명령을 수신 받으면, 임시 저장 레지스터(TR) 또는 디버깅 스택 포인터(ISP)가 지정하는 스택에 저장된 레지스터의 데이터를 호스트 컴퓨터로 전송하고, 사용자는 전송된 데이터를 분석하여 올바른 값이 출력되었는지를 판단하고, 분석 결과 올바른 데이터가 아니면 응용 프로그램이 올바르게 작성되지 않은 것이므로 응용 프로그램을 수정한다.

【보정대상항목】 식별번호 55

【보정방법】 정정

【보정내용】

리셋 신호(RESET)는 중앙처리장치에 최초 전압 인가시 활성화되거나 리셋 데이터 비교부(1000)에 의해서 데이터 통신부(600)를 통해 호스트 컴퓨터로부터 입력되는 디버깅 데이터(DEB2)와 리셋 데이터 저장부(1100)에 저장된 리셋 데이터를 비교하여 동일하면 중앙처리장치를 초기화시킨다. 이는 디버깅 할 응용프로그램의 수행 중 중앙처리장치가 복구 불가능한 상태, 예를 들어 무한루프 등의 상태에 있을 경우 중앙처리장치를 초기화시켜 문제를 해결한다.

【보정대상항목】 식별번호 56

【보정방법】 정정

【보정내용】

응용 프로그램이 일정 시간 이상 계속해서 실행되는 경우, 즉 응용 프로그램이 무한 루프에 있는 경우 데이터 통신부(600)를 통해 기준 데이터 저장부(900)에 저장된 기준 데이터와 동일한 디버깅 데이터(DEB1)를 송신하면, 기준 데이터 비교부(800)는 기준 데이터 저장부(900)에 저장된 기준 데이터와 데이터 통신부(600)를 통해 받은 디버깅 데이터(DEB1)를 비교하여 같으면 강제적으로 중앙처리장치를 프로그램 테스트 및 디버깅 상태인 디버깅 서비스 모드로 진입시켜 무한 루프에서 벗어나서 무한 루프가 발생된 부분의 응용 프로그램을 디버깅 할 수 있다.

【보정대상항목】 식별번호 57

【보정방법】 정정

【보정내용】

도 5c에 도시된 바와 같이 디버깅 모드 초기화 프로그램(PROG3, PROG6) 및 디버깅 서비스 프로그램(PROG5, PROG7)은 중앙처리장치를 초기화 및 동작시키기 위한 서비스 프로그램이 저장된 메모리(80)와 별도의 메모리(90)에 동시에 저장될 수 있다. 이를 위해 제어유니트(300)의 입력신호로 메모리 구분 신호(OSIROM)가 필요하다. 즉 리셋 신호(RESET)에 의해 제어유니트(300)가 중앙처리장치를 초기화시키고 디버깅 모드 진입 신호(OSI)를 검사하여 활성화되었으면, 메모리 구분 신호(OSIROM)를 검사하여 활성화되었으면, 프로그램 카운터(PC)에는 응용 프로그램의 테스트 및 디버깅을 위해 메모리(90)에

저장된 디버깅 모드 초기화 프로그램(PROG6)의 시작 주소인 디버깅 모드 초기화 벡터2가 저장되어야 한다. 그리고 브레이크 레지스터(BR)의 데이터 값과 중지데이터가 동일할 경우 비교부(700)는 두 값이 동일함을 제어유니트(300)에 알려주고 제어유니트(300)는 프로그램 카운터(PC)에 메모리(90)에 저장된 디버깅 서비스 프로그램(PROG7)의 시작 주소인 디버깅 서비스 벡터2를 저장한다.

【보정대상항목】 식별번호 58

【보정방법】 정정

【보정내용】

만약에 메모리 구분 신호(OSIROM)가 비활성화되면, 디버깅 모드 초기화 벡터1이 디버깅 모드 초기화 프로그램(PROG3)을 수행하기 위한 시작 주소이고 디버깅 서비스 벡터1이 디버깅 서비스 프로그램(PROG5)을 수행하기 위한 시작 주소가 된다.

【보정대상항목】 식별번호 60

【보정방법】 정정

【보정내용】

또한 본 발명의 프로그램 테스트 및 디버깅이 용이한 중앙처리장치는 리셋 신호(RESET)에 의한 중앙처리장치의 초기화 후 디버깅 모드 진입 신호(OSI)에 의해 최초 디버깅 모드로 전환되거나 또는 중앙처리장치가 응용 프로그램을 수행하는 중에 호스트 컴퓨터로부터 데이터 통신부(600)를 통해 기준 데이터 저장부(900)에 저장된 기준데이터와 동일한 디버깅 데이터(DEB1)의 입력에 의하여 제어유니트(300)에 의해 강제적으로 최초 디버깅 모드로 전환되어 디버깅 초기화 프로그램과 디버깅 서비스 프로그램으로 이루어

진 디버깅 프로그램이 수행되면 디버깅 프로그램에 의해 호스트 컴퓨터에서 브레이크 레지스터(BR)의 값을 설정할 수 있다. 설정된 브레이크 레지스터(BR)의 데이터는 데이터 통신부(600)를 통해 브레이크 레지스터(BR)에 저장되어 중앙처리장치가 응용 프로그램 수행 중 브레이크 레지스터(BR)에 저장된 값(IBR)과 중지데이터(BD)가 동일하면 디버깅 모드 진입 신호(OSI)의 활성화 여부에 관계없이 항상 중앙처리장치를 디버깅 모드로 진입하게 한다. 즉, 브레이크 레지스터(BR)에 저장된 값(IBR)과 중지데이터(BD)가 동일하면 제어유니트(300)는 상태 레지스터(SR)의 디버깅 모드 플래그(OSIM)를 활성화시키고, 프로그램 카운터(PC)에는 디버깅 프로그램을 수행하기 위한 시작 주소를 적재하여 중앙처리장치는 디버깅 프로그램을 수행하도록 하여 디버깅 모드로 전환시키고, 디버깅 프로그램에 의해 데이터 통신부(600)를 통해 호스트 컴퓨터의 명령, 예를 들어 디버깅 프로그램에서 사용할 메모리의 초기화, 디버깅 스택 포인터의 설정, 기준데이터의 설정이나 변경, 중앙처리장치를 초기화시키기 위한 리셋데이터의 설정이나 변경, 브레이크 레지스터(BR) 값의 변경, 범용 레지스터 파일(100) 및 특수 레지스터 파일(200)의 레지스터들에 저장된 데이터의 판독(READ) 또는 기록(WRITE) 및 메모리에 저장된 데이터들의 판독 또는 기록 등의 명령을 수행하여 응용 프로그램이 올바르게 작성되었는지를 판단하고, 잘못 작성되었으면 이를 수정하여 바로 잡는다.

【보정대상항목】 식별번호 61

【보정방법】 삭제

【보정대상항목】 식별번호 62

【보정방법】 삭제

【보정대상항목】 식별번호 63

【보정방법】 삭제

【보정대상항목】 식별번호 64

【보정방법】 삭제

【보정대상항목】 식별번호 65

【보정방법】 삭제

【보정대상항목】 식별번호 66

【보정방법】 삭제

【보정대상항목】 식별번호 67

【보정방법】 삭제

【보정대상항목】 식별번호 68

【보정방법】 삭제

【보정대상항목】 청구항 1

【보정방법】 정정

【보정내용】

데이터 및 어드레스 연산에 필요한 데이터를 일시적으로 저장하는 범용 레지스터 파일, 프로그램이 저장된 메모리의 주소를 기억하는 프로그램 카운터와 중앙처리장치의 상태를 나타내는 상태 레지스터 및 브레이크 레지스터로 구성된 특수 레지스터 파일, 상기의 범용 레지스터 파일 및 특수 레지스터 파일을 연결하는 내부 버스, 상기의 내부 버스에 연결되어 중앙처리장치 내외부에 필요한 제어신호를 출력하는 제어유니트로 구성된 중앙처리장치에 있어서,

호스트 컴퓨터와 데이터 통신을 위한 데이터 통신수단;

중앙처리장치의 동작모드가 일반 동작 상태를 나타내는 일반 동작 모드인지 디버깅 상태를 나타내는 디버깅 모드인지를 나타내는 플래그를 구비한 상태 레지스터;

디버깅 프로그램의 데이터를 저장하는 스택 메모리를 지정하는 스택 포인트로 사용되는 디버깅 스택 포인터 레지스터; 및

상기의 브레이크 레지스터에 저장된 값과 중지데이터를 비교하여 동일하면 상기의 중앙처리장치를 디버깅 모드로 진입시키는 비교수단을 구비하여 상기의 브레이크 레지스터에 저장된 값과 상기의 중지 데이터가 일치하면 상기의 중앙처리장치를 디버깅 모드로

전환시키고 상기의 상태 레지스터의 플래그가 디버깅 모드를 나타내는 값을 가지고 상기의 프로그램 카운터에 디버깅 프로그램을 수행하기 위한 시작 주소를 적재하여 디버깅 프로그램을 수행하여 상기의 데이터 통신수단을 통해 호스트 컴퓨터로부터의 명령에 따라 디버깅을 수행하는 것을 특징으로 하는 프로그램 테스트 및 디버깅이 용이한 중앙처리장치.

【보정대상항목】 청구항 2

【보정방법】 정정

【보정내용】

제1항에 있어서, 상기의 프로그램 테스트 및 디버깅이 용이한 중앙처리장치는

리셋 데이터가 저장된 리셋 데이터 저장수단; 및

상기의 데이터 통신수단을 통해 입력받은 데이터와 상기의 리셋 데이터 저장수단에 저장된 리셋 데이터를 비교하여 상기의 데이터 통신수단을 통해 입력받은 데이터와 리셋 데이터가 동일하면 제어유니트가 중앙처리장치를 초기화하도록 지시하는 리셋 데이터 비교수단을 더 구비한 것을 특징으로 하는 프로그램 테스트 및 디버깅이 용이한 중앙처리장치.

【보정대상항목】 청구항 11

【보정방법】 정정

【보정내용】

제1항에 있어서, 상기의 제어유니트는 디버깅 메모리 선택신호를 수신하여 상기의 브레이크 레지스터에 저장된 값과 상기의 중지데이터가 동일할 때 상기의 디버깅 메모리

선택신호에 따라 상기의 프로그램 카운터에 서로 다른 디버깅 프로그램을 수행하기 위한 주소를 적재하는 것을 특징으로 하는 프로그램 테스트 및 디버깅이 용이한 중앙처리장치.

【보정대상항목】 청구항 12

【보정방법】 정정

【보정내용】

제1항에 있어서, 상기 디버깅 프로그램에서 사용하는 데이터 값을 저장하는 데이터 저장 메모리를 더 구비하여 일반 프로그램에서 사용하는 데이터 값을 저장하는 데이터 저장 메모리와 분리한 것을 특징으로 하는 프로그램 테스트 및 디버깅이 용이한 중앙처리장치.

【보정대상항목】 청구항 13

【보정방법】 정정

【보정내용】

제1항에 있어서, 상기 디버깅 프로그램을 저장하는 메모리를 더 구비하여 일반 프로그램이 저장된 메모리와 분리한 것을 특징으로 하는 프로그램 테스트 및 디버깅이 용이한 중앙처리장치.

【보정대상항목】 청구항 14

【보정방법】 정정

【보정내용】

제1항에 있어서, 상기의 데이터 통신 수단을 통해 상기의 호스트 컴퓨터로부터 테스트 및 디버깅 할 응용 프로그램을 다운로드 받는 것을 특징으로 하는 프로그램 테스트 및 디버깅이 용이한 중앙처리장치.

【보정대상항목】 청구항 15

【보정방법】 삭제

【보정대상항목】 청구항 18

【보정방법】 정정

【보정내용】

데이터 및 어드레스 연산에 필요한 데이터를 일시적으로 저장하는 범용 레지스터 파일, 프로그램이 저장된 메모리의 주소를 기억하는 프로그램 카운터와 중앙처리장치의 상태를 나타내는 상태 레지스터 및 브레이크 레지스터로 구성된 특수 레지스터 파일 및 상기의 범용 레지스터 파일 및 특수 레지스터 파일을 연결하는 내부 버스로 구성된 중앙 처리장치에 있어서,

호스트 컴퓨터와 데이터 통신을 위한 데이터 통신수단;

중앙처리장치의 동작모드가 일반 동작 상태를 나타내는 일반 동작 모드인지 디버깅 상태를 나타내는 디버깅 모드인지를 나타내는 플래그를 구비한 상태 레지스터;

디버깅 초기화 프로그램 및 디버깅 서비스 프로그램의 데이터를 저장하는 스택 메모리를 지정하는 디버깅 스택 포인터 레지스터;

리셋 신호에 의해 상기의 중앙처리장치를 초기화하고 디버깅 모드 진입 신호를 수신하여 디버깅 모드 진입 신호가 활성화되면 상기 프로그램 카운터에 디버깅 초기화 프로그램을 수행하기 위한 시작 주소를 적재하여 상기의 중앙처리장치를 디버깅 초기화 모드로 전환시키고, 상기 상태 레지스터의 플래그를 디버깅 모드를 나타내는 값으로 설정하고, 상기 내부 버스에 연결되어 중앙처리장치의 내외부에 필요한 제어신호를 출력하는 제어유니트; 및

상기 브레이크 레지스터에 저장된 값과 중지데이터를 비교하여 동일하면 상기 중앙처리장치를 디버깅 서비스 모드로 진입시키는 비교수단을 구비하여 상기 브레이크 레지스터에 저장된 값과 상기 중지데이터가 동일하면 상기의 중앙처리장치를 디버깅 모드로 전환시키고 상기의 상태 레지스터의 플래그가 디버깅 서비스 모드를 나타내는 값을 가지고 상기의 프로그램 카운터에 디버깅 서비스 프로그램을 수행하기 위한 시작 주소를 적재하여 디버깅 서비스 프로그램을 수행하여 상기의 데이터 통신수단을 통해 호스트 컴퓨터로부터의 명령에 따라 디버깅을 수행하는 것을 특징으로 하는 프로그램 테스트 및 디버깅이 용이한 중앙처리장치.

【보정대상항목】 청구항 19

【보정방법】 정정

【보정내용】

제18항에 있어서, 상기의 프로그램 테스트 및 디버깅이 용이한 중앙처리장치는

리셋 데이터가 저장된 리셋 데이터 저장수단; 및

상기의 데이터 통신수단을 통해 입력받은 데이터와 상기의 리셋 데이터 저장수단에 저장된 리셋 데이터를 비교하여 상기의 데이터 통신수단을 통해 입력받은 데이터와 리셋 데이터가 같으면 상기의 중앙처리장치를 초기화하는 리셋 데이터 비교수단을 구비한 것을 특징으로 하는 프로그램 테스트 및 디버깅이 용이한 중앙처리장치.

【보정대상항목】 청구항 20

【보정방법】 정정

【보정내용】

제18항에 있어서, 상기 디버깅 프로그램에서 사용하는 데이터 값을 저장하는 데이터 저장 메모리를 더 구비하여 일반 프로그램에서 사용하는 데이터 값을 저장하는 데이터 저장 메모리와 분리한 것을 특징으로 하는 프로그램 테스트 및 디버깅이 용이한 중앙처리장치.

【보정대상항목】 청구항 21

【보정방법】 정정

【보정내용】

제18항에 있어서, 상기 디버깅 프로그램을 저장하는 메모리를 더 구비하여 일반 프로그램이 저장된 메모리와 분리한 것을 특징으로 하는 프로그램 테스트 및 디버깅이 용이한 중앙처리장치.

【보정대상항목】 청구항 23

【보정방법】 삭제

【보정대상항목】 청구항 26

【보정방법】 삭제

【보정대상항목】 청구항 27

【보정방법】 삭제

【보정대상항목】 청구항 28

【보정방법】 정정

【보정내용】

제18항에 있어서, 상기의 제어유니트는 디버깅 메모리 선택신호를 수신하여 상기의 브레이크 레지스터에 저장된 값과 상기의 중지데이터가 동일할 때 상기의 디버깅 메모

리 선택신호에 따라 상기의 프로그램 카운터에 디버깅 프로그램을 수행하기 위한 서로 다른 주소를 적재하는 것을 특징으로 하는 프로그램 테스트 및 디버깅이 용이한 중앙처리장치.

【보정대상항목】 청구항 29

【보정방법】 삭제

【보정대상항목】 청구항 30

【보정방법】 삭제

【보정대상항목】 청구항 31

【보정방법】 삭제

【보정대상항목】 청구항 32

【보정방법】 삭제

【보정대상항목】 청구항 33

【보정방법】 삭제

【보정대상항목】 청구항 34

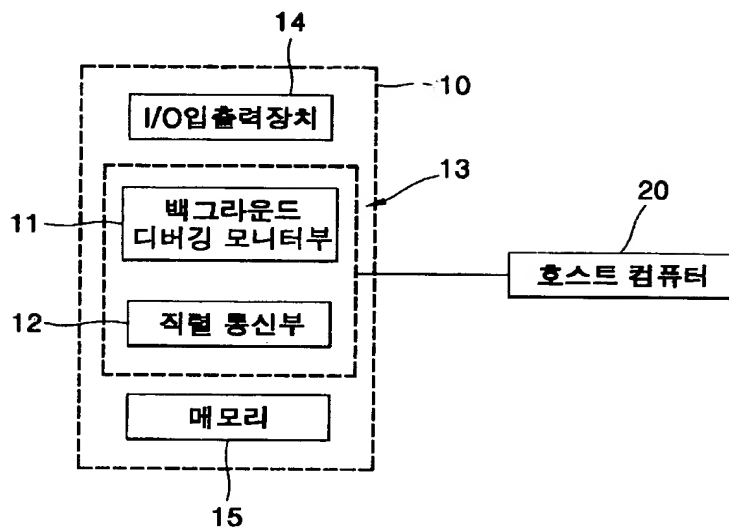
【보정방법】 삭제

【보정대상항목】 도 2

【보정방법】 정정

【보정내용】

【도 2】

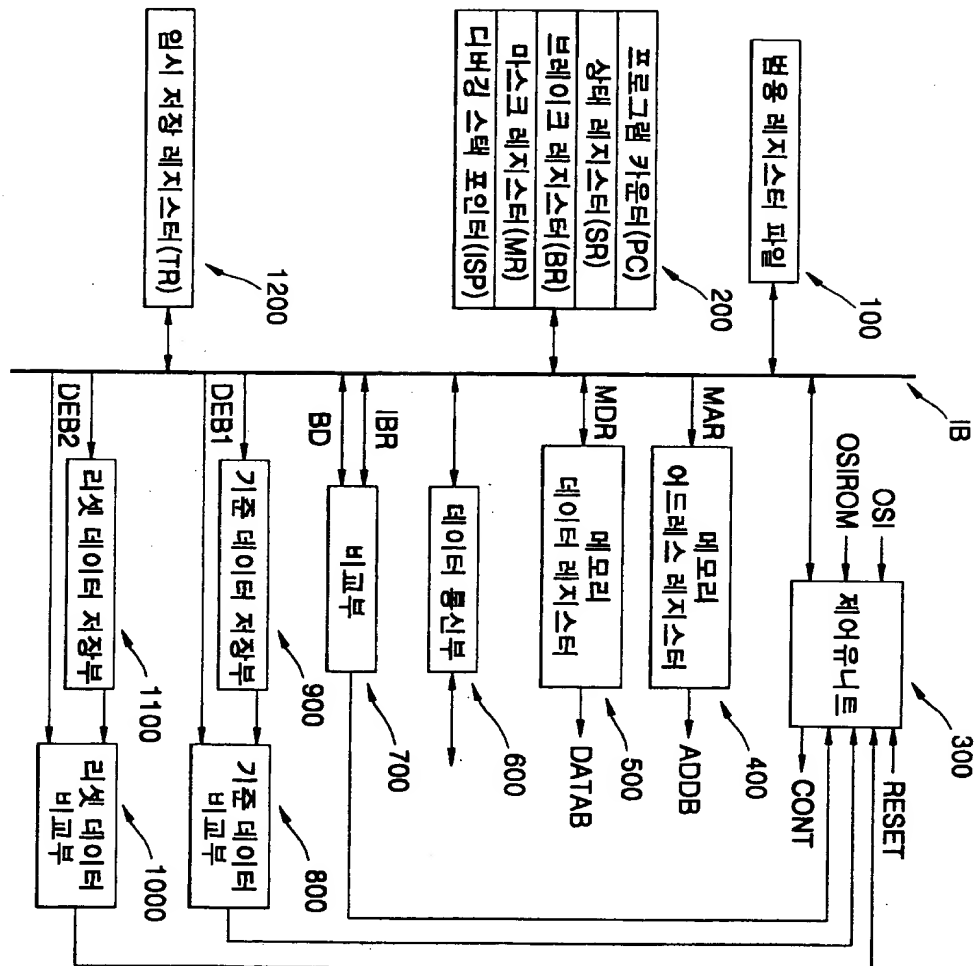


【보정대상항목】 도 3

【보정방법】 정정

【보정내용】

【도 3】

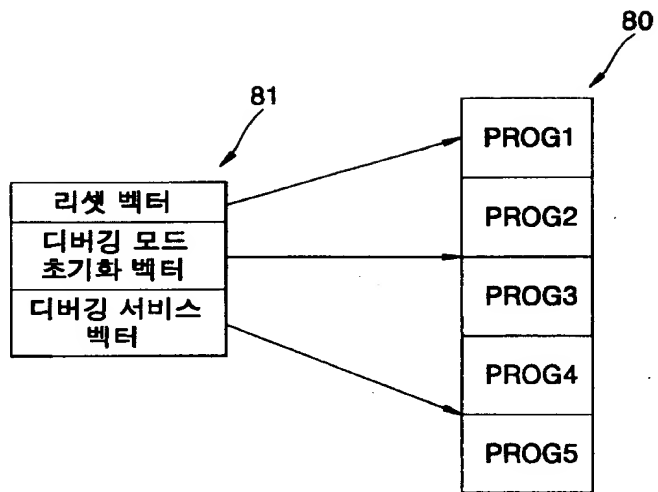


【보정대상항목】 도 5a

【보정방법】 정정

【보정내용】

【도 5a】

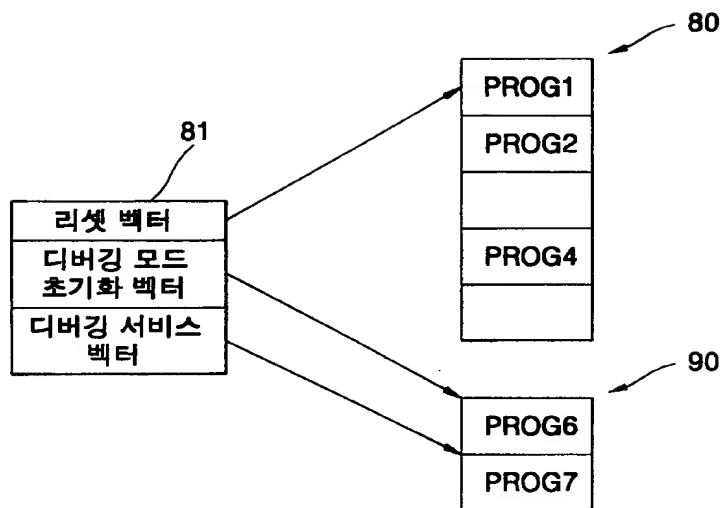


【보정대상항목】 도 5b

【보정방법】 정정

【보정내용】

【도 5b】

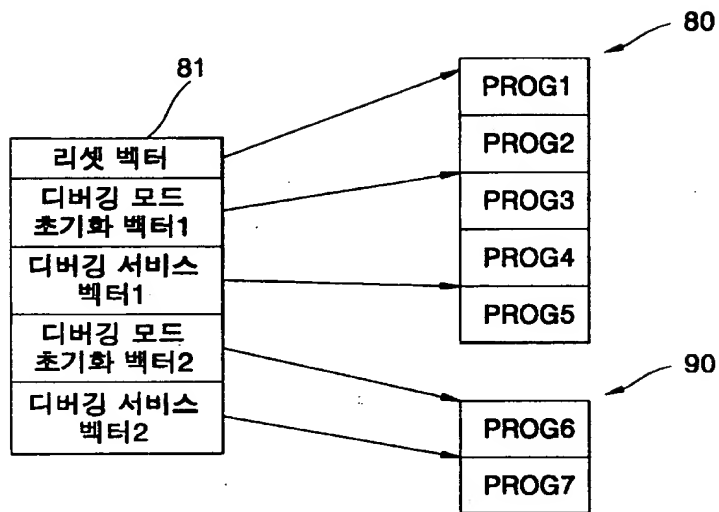


【보정대상항목】 도 5c

【보정방법】 정정

【보정내용】

【도 5c】



【보정대상항목】 도 6

【보정방법】 삭제

【보정대상항목】 도 7

【보정방법】 삭제